SWT
SOFTWARE TOOLS

English User Manual

Copyright © (1990-2022) SW-Tools ApS
Duevej 23
DK-2680 Solrød Strand
Denmark
Phone: **+45) 33 33 05 56**
Mail: **swtools@swtools.com**
www: **www.swtools.com**

WIN-TE

22/11/01 / 2022-09-01 008.384

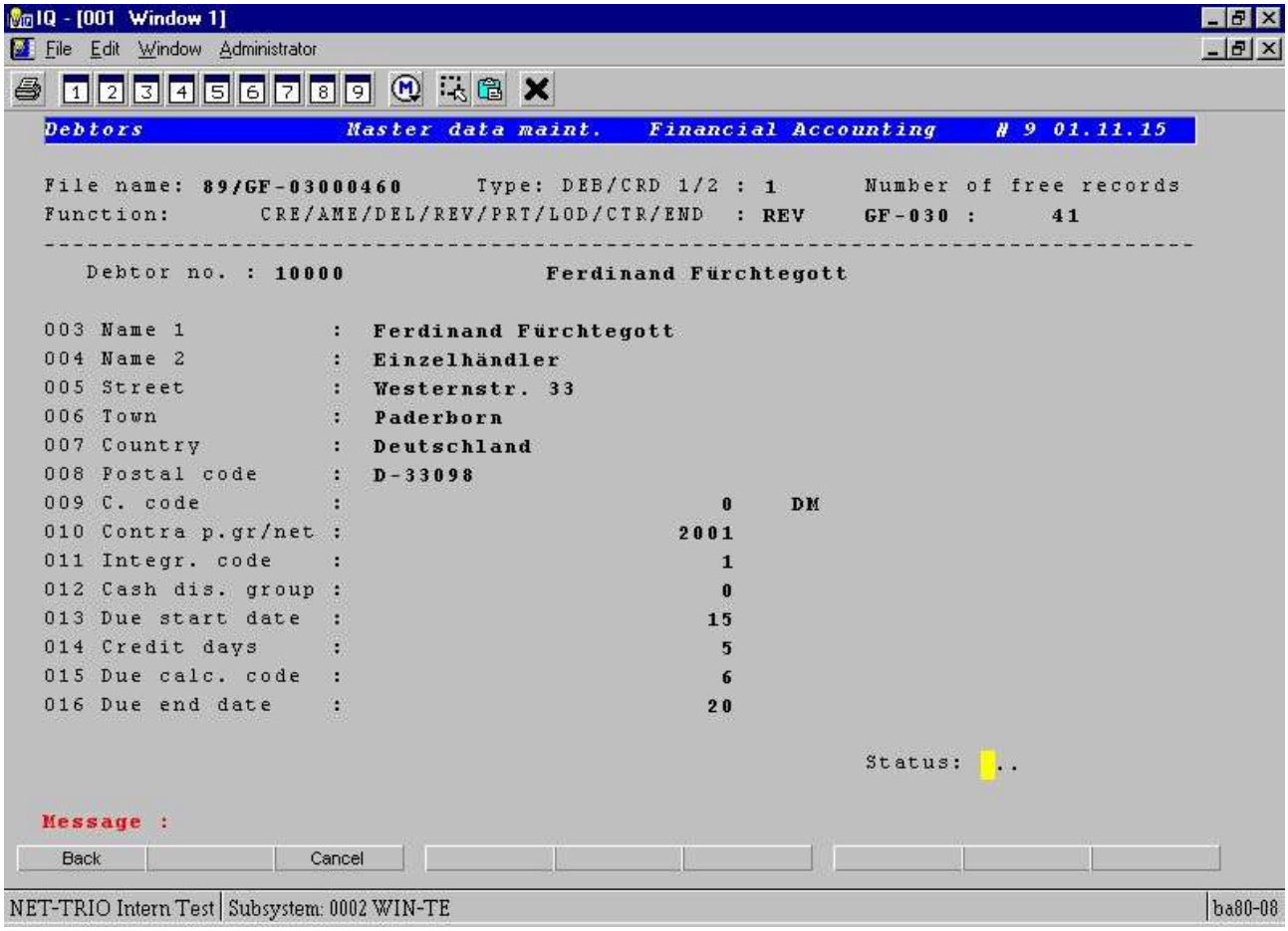
Contents

Contents 2
1. Preface 4
1.1. Enhancements 6
1.2. Supported systems 7
1.3. License 8
1.4. Application requirements 9
1.5. NET-TRIO - Intranet/Internet enable your application 10
1.6. Examples 11
2. The WIN-TE environment 12
2.1. The WIN-TE applications 13
3. How to extend an application step by step 14
3.1. Login to the system 15
3.2. Start the application 16
3.3. Unique mask number 17
3.4. Automatic recognition of input fields 18
3.5. Defining pop-up menus for 3-letter commands 19
3.6. Defining a lookup field 22
3.6.1. Background change equals new mask 23
3.6.2. Defining the IQ query on debtor 24
3.6.3. Defining the action on the debtor field 27
3.7. Defining input fields manually 30
3.8. Defining display fields manually 31
3.9. Defining button fields 32
3.9.1. Copy application text and call window application 37
4. Preferences 39
4.1. The preference dialog 40
4.2. Profiles 41
4.2.1. How to create a new profile 42
4.2.2. How to delete a profile 43
4.3. Connection 44
4.3.1. Server 45
4.3.2. Service port 46
4.3.3. Terminal type 47
4.3.4. Screen number 48
4.3.5. Logo 49
4.3.6. Login dialog 50
4.3.7. Flags 51
4.4. Terminal 53
4.4.1. Columns 54
4.4.2. Lines 55
4.4.3. Message line 56
4.4.4. Line space 57
4.4.5. Telnet definition 58
4.5. Keyboard 59
4.5.1. Timer 60
4.5.2. Accelerator 61
4.5.3. Cursor 62
4.5.3.1. Animation 63
4.5.4. Key definition 64
4.6. Background color 65
4.7. Pen color 66
4.8. Text color 67

4.9. Font 68
4.9.1. 3D effect 69
4.10. Mask 70
4.10.1. Mask definition 71
4.10.2. Mask number X 72
4.10.3. Mask number Y 73
4.10.4. Mask compare length 74
4.10.5. Mask number definition 75
4.10.6. Forground definition 76
4.10.7. Background definition 77
4.11. Logging 78
4.11.1. Log file 79
4.12. Open Basic NT 80
4.13. NetBasic 81
4.14. Global Text 82
4.15. Profile Text 83
5. Printing 84
5.1. Printing using scripts 85
5.1.1. Output via the screen (standard output) 86
5.1.2. Output via FTP transfer 87
5.2. Cross Basic 88
5.2.1. Required environment variables 89
5.2.2. SW-Tools WIN-TE Print Server 90
5.2.3. Device assignment 91
5.2.4. SWOI application parameters 92
5.3. Codetable conversion and filter 93
5.3.1. Codetable 94
5.3.2. Filter 95
5.4. Extended print control 96
5.5. Archiving 97
5.6. Forms 98
6. Functions 99
6.1. TelnetAllocate - Allocate Telnet object 100
6.2. TelnetFree - Free telnet object 101
6.3. TelnetConnect - Connect Telnet 102
6.4. TelnetDisconnect - Disconnect Telnet 103
6.5. TelnetPut - Put text to Telnet 104
6.6. TelnetNotify - Notify when text received 105
6.7. TelnetKey - Put key sequence to Telnet 106
6.8. TelnetGetnum - Get Telnet numeric variable 107
6.9. TelnetGettxt - Get Telnet text variable 108
6.10. TelnetSaveScreen - Save Telnet screen content 110
6.11. TelnetActivate - Activate focus on Telnet 111
6.12. TelnetPref - Preferences 112
6.13. TelnetSetnum - Set Telnet numeric variable 113
6.14. TelnetSettxt - Set Telnet text variable 114
6.15. TelnetFileFilter - Filter file according to printer filter 115
Figure list 116
Index 117

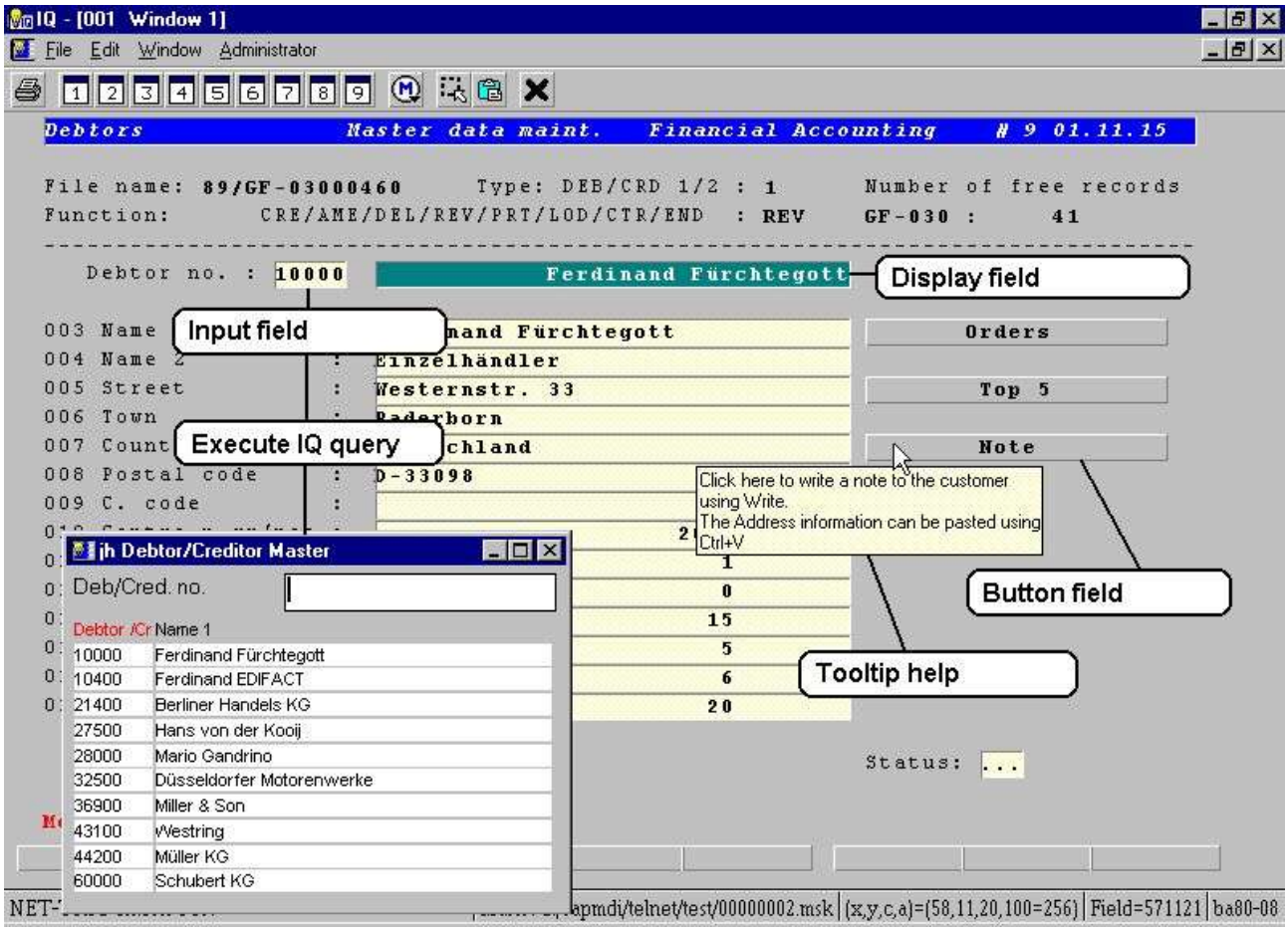
1. Preface

SW-Tools WIN-TE enables you to enhance **ANY** character based application **without** modifications.
Throughout the manual will refer to a TELNET object which is the foundation of SW-Tools WIN-TE.
As in a commercial we start by presenting the before picture



1. Application appearance before using WIN-TE

and the after picture of a character based application:



2. Application appearance after using WIN-TE

1.1. Enhancements

The possible enhancements are:

- Mask definition to extend mask with input, display and button fields
- Data grids by click on a field
- Pop-up menus by click on a field
- Direct execution of any TRIO Report/Query/Data Entry program by click on a field
- Direct execution of any Windows application by click on a field including copy of any text from the application screen
- Tooltip on any defined input, display and button field
- 3D effect on input, display and button fields
- Display of pictures, charts and other graphical material through SW-Tools TRIO
- Play video, sound, presentations through OLE in SW-Tools TRIO
- Printing to screen, html, pdf and other printers supported by SW-Tools and Windows

and

- Any other enhancement you may think of which may be defined in SW-Tools TRIO
- Any other enhancement you may think of using a Windows application

Apart from the enhancements you may accomplish with WIN-TE it also means that

- No modification is needed in the existing application environment
- No additional programming is needed in the existing application environment

1.2. Supported systems

WIN-TE do not have any requirement to the host that currently runs the character based application. All that is required is that the application can be run in a terminal emulator using the system service named **telnet**. Therefore WIN-TE support

- **Any UNIX system**
- **OpenBasic**
- **NetBasic (Not ready for release**)

1.3. License

The TELNET object is an add-on object to the SW-Tools TRIO product. It is licensed with a separate product code 0107.

1.4. Application requirements

All that is required is that any screen/mask within the application has got a mask number located at a location on the screen which is unique. If not, it is also possible to let the object generate the mask number according to the background content of the screen or any text according to a fixed screen position, for example an application name in the heading line.

1.5. NET-TRIO - Intranet/Internet enable your application

With the TELNET object and all the enhancement possibilities you will first of all be able to enhance you application.
But wouldn't it be nice to also have access to this from a browser in your intranet or from the internet ?
It is possible because SW-Tools NET-TRIO supports any defined IQ query and thus having the TELNET object as part of IQ, NET-TRIO support your character based application.
There are some of the enhancement functionality that is not yet released for NET-TRIO. These are:

- Tooltip help on fields

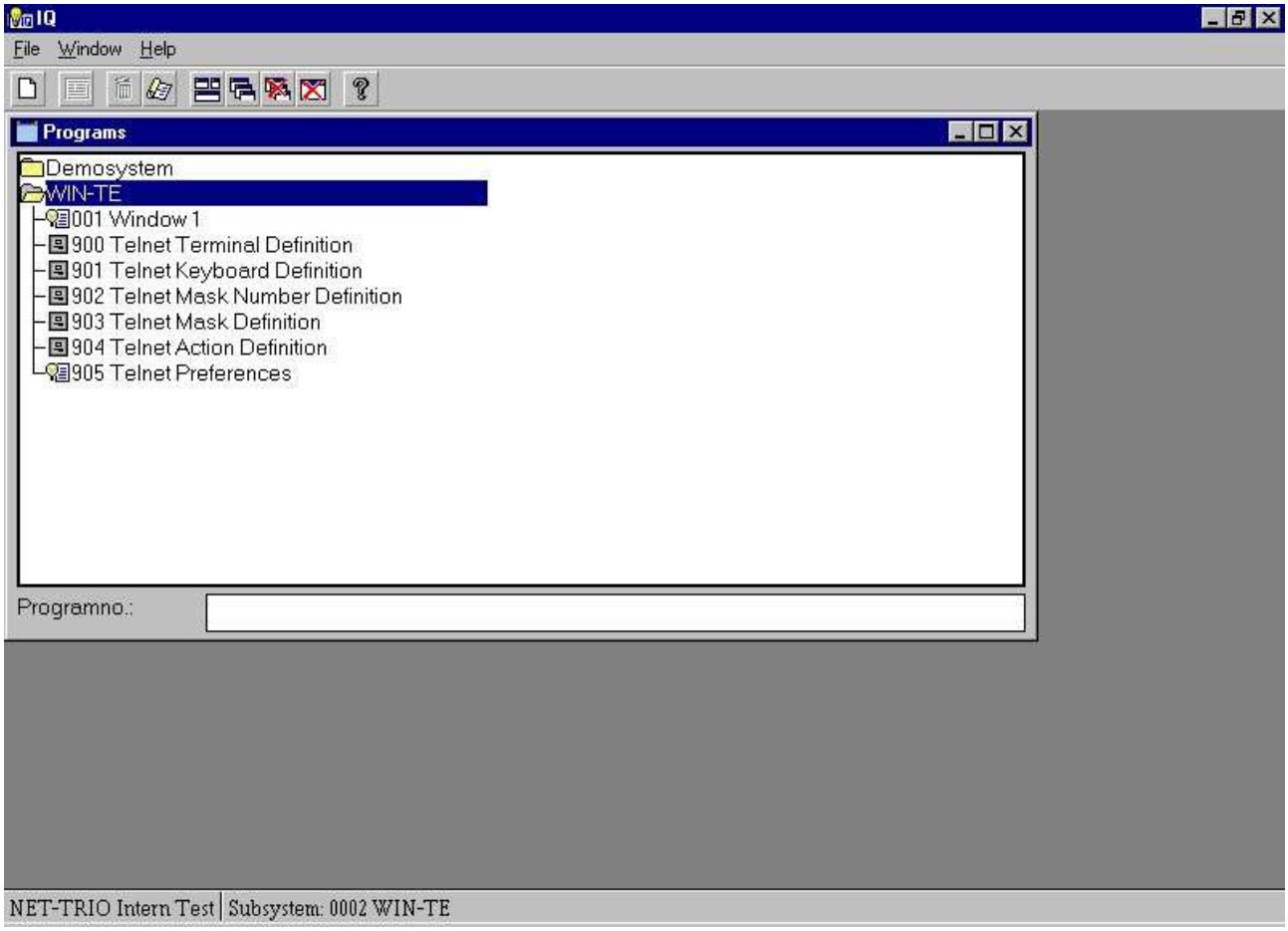
1.6. Examples

It would be almost impossible to write a manual with examples that covers any type of character based application. Therefore we have decided to use the character based Business Basic application named **COMET**.
Our system setup is a PC with Windows NT, CrossBasic NT and the Business Basic application COMET.
This could just as easely have been any application written for LINUX, SCO UNIX, Siemens RM Series with SINIX or any other UNIX system.

2. The WIN-TE environment

2.1. The WIN-TE applications

When you have installed SW-Tools IQ and WIN-TE you should have the following IQ and DATAMASTER applications available.



3. The WIN-TE applications

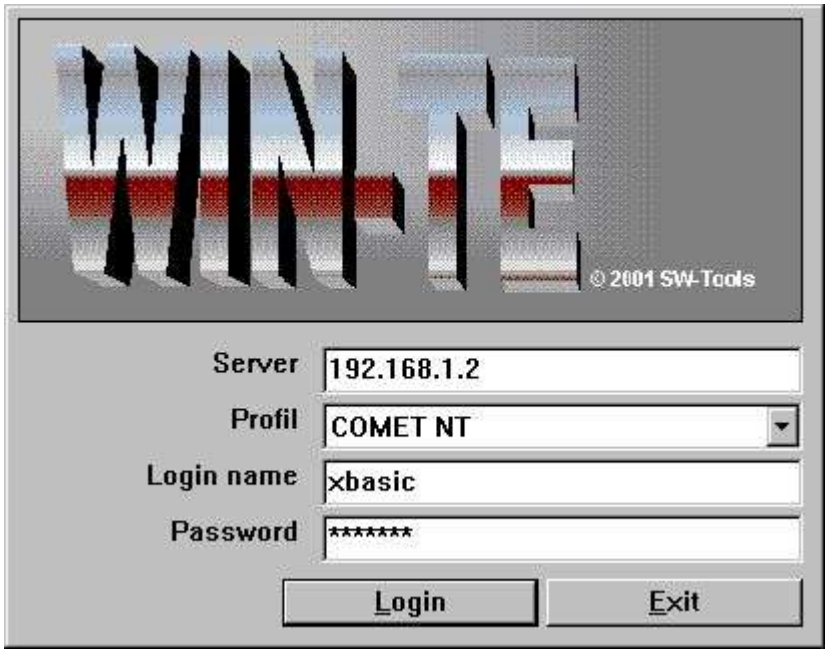
The applications 900-905 are all defined with IQ/DATAMASTER and is only used when enhancements are to be defined and NOT when the user operates the system.

3. How to extend an application step by step

We have limited our example to be based on the application screen for **Debtors Master data maintenance**.

3.1. Login to the system

The first step is to start the IQ application 1 named "Window 1" and login to your system.



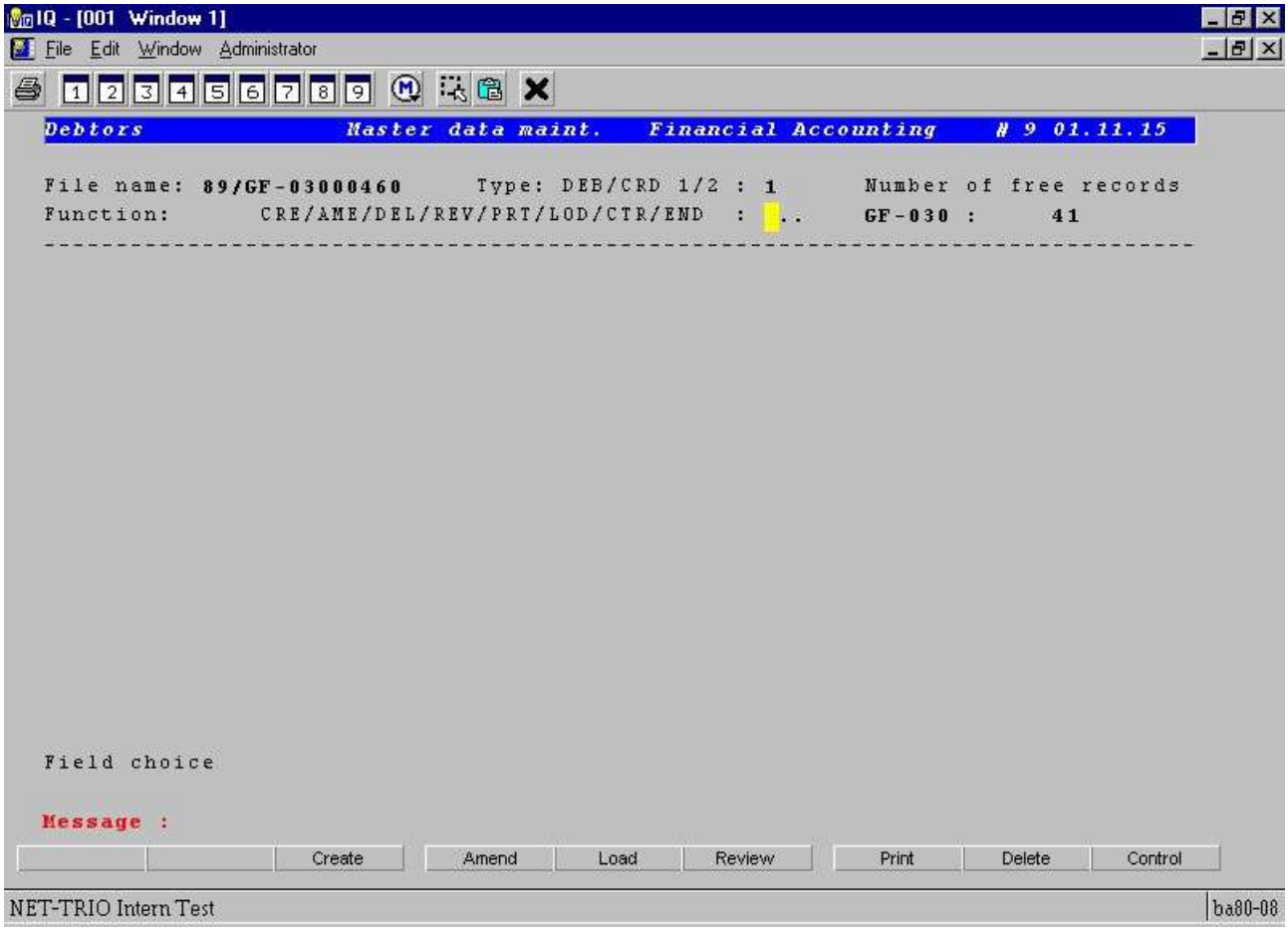
4. Step 1 - Login to the system

Please remember to select the correct profile for your system. By standard installation the following profiles are available:

- **COMET NT**
- **COMET RM**
- **COMET SCO**
- **Open Basic**
- **NetBasic**
- **Standard (For standard VT220 connections)**

3.2. Start the application

The second step is to start the application where enhancement is required.



5. Step 2 - Started Debtors Master data maintenance

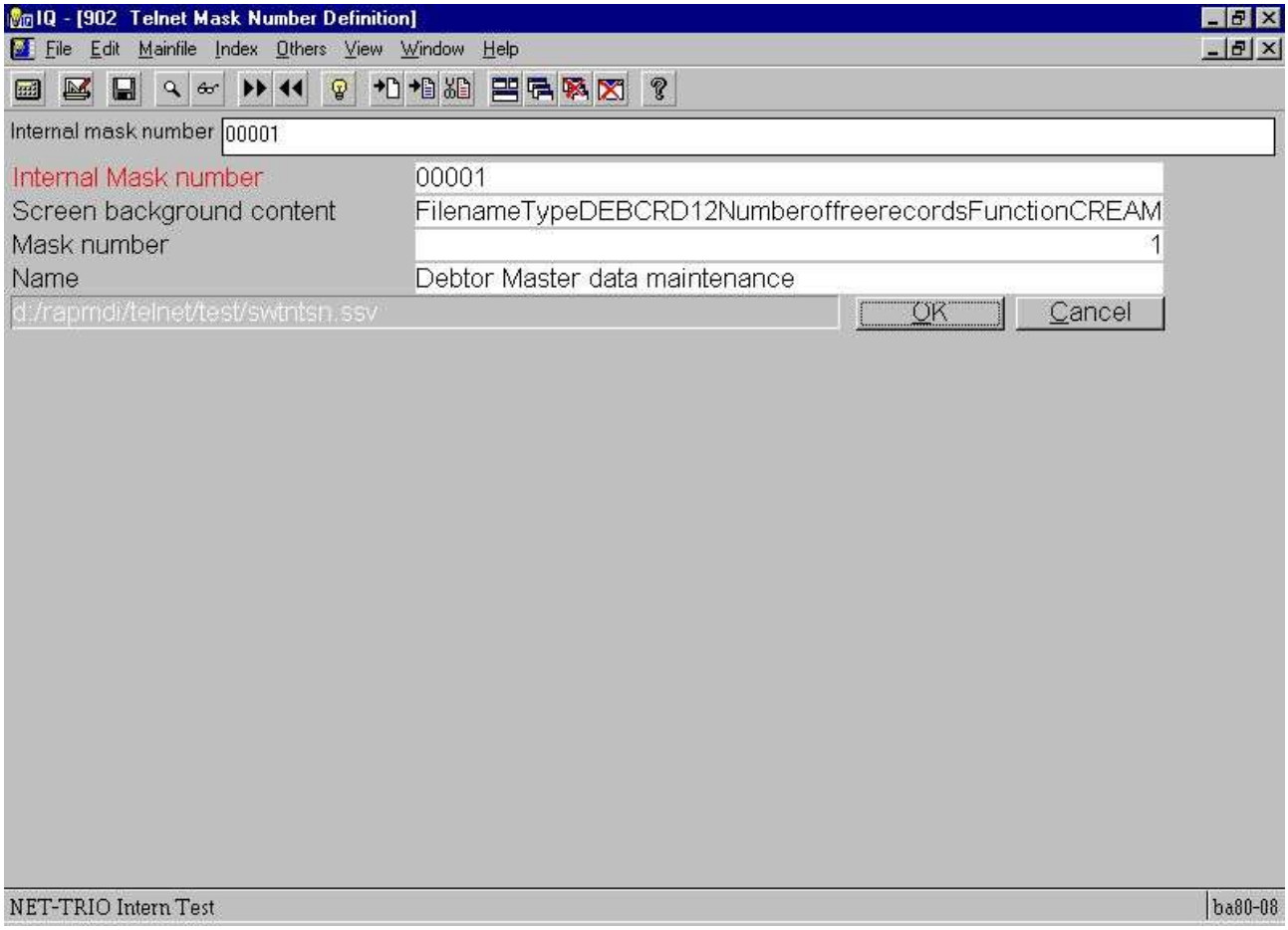
3.3. Unique mask number

Now because your sample application do not have a unique mask number it can be generated according to the background text.
This is accomplished by the following procedure:

- Clicking on the first column on the first line, e.g. the topmost left position of the application window.

When done the emulator will call the DATAMASTER application 902. It will provide 4 fields:

- Internal mask number
- Screen background content
- Mask number
- Name

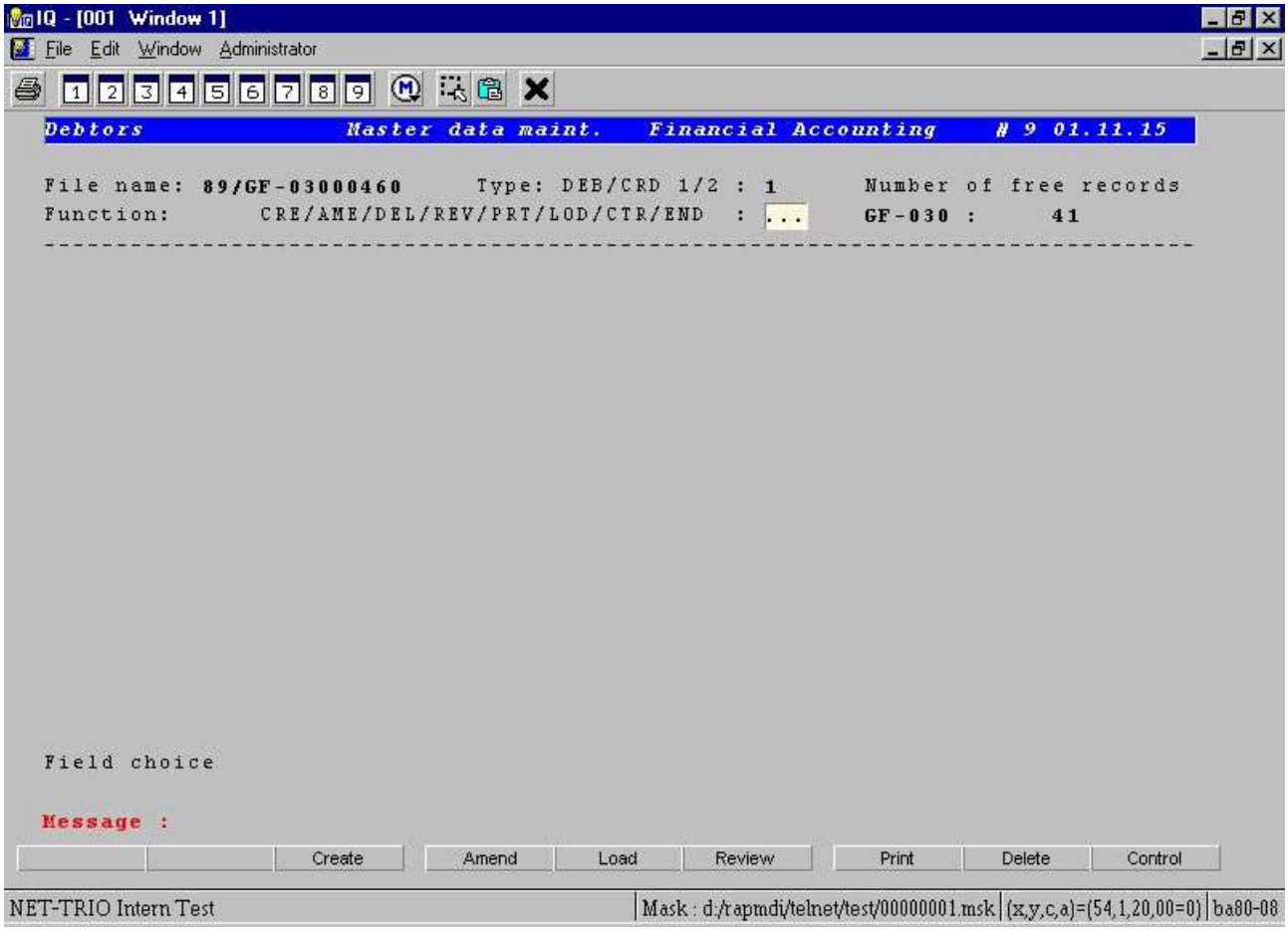


6. Step 3 - Definition of a unique mask number

The mask number is initially set to the internal mask number but you may modify it. This may be useful if multiple backgrounds are used for the same mask.
The name field is useful to give the mask a meaningful name, e.g. **Debtors Master data maintenance**.
Once you save the mask definition by selecting **OK** and close the DATAMASTER application you will return to the emulation screen and the new mask number is active.
As soon as a unique mask number is present you may continue you extension of the application appearance and functionality.

3.4. Automatic recognition of input fields

In this application all input fields starts up with dots. The preference setting have an option for automatic recognition of such a field as soon as a unique mask number is present. Therefore the first field is automatically inserted in the new mask definition.



7. Step 4 - Automatic recognition of (dotted) input fields

3.5. Defining pop-up menus for 3-letter commands

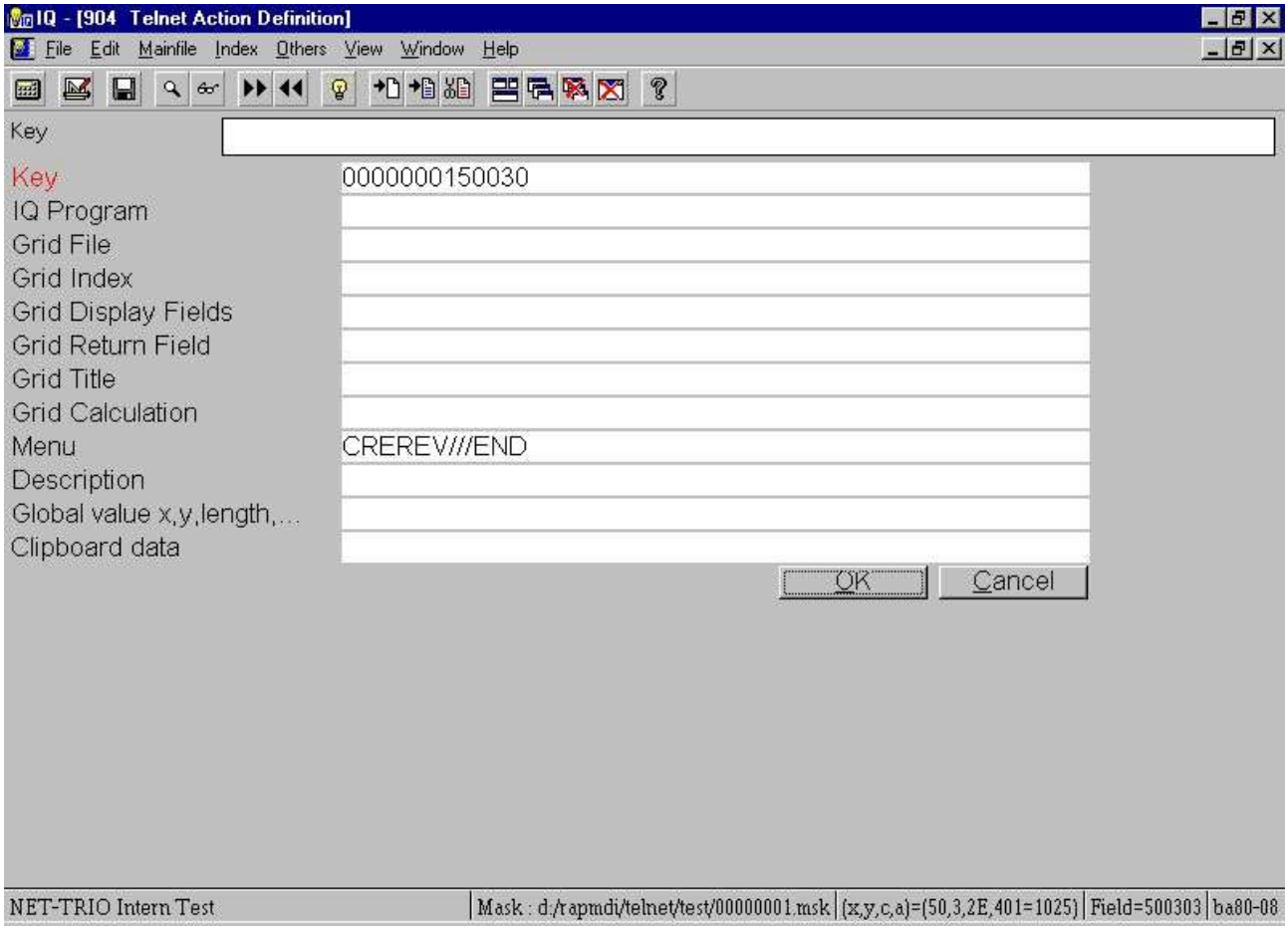
The first field is normally used to select which function the user want to activate on the debtor, e.g a 3 letter command where

CRE is Create a new customer
REV is Review an existing customer
...
END is Exit the application

You will now enhance the application by providing the user with a pop-up menu where each menu item is the meaningful description of the command rather than the command itself.
You need to enhance the field with an action when the left mouse button is clicked. You may define such an action by the following procedure:

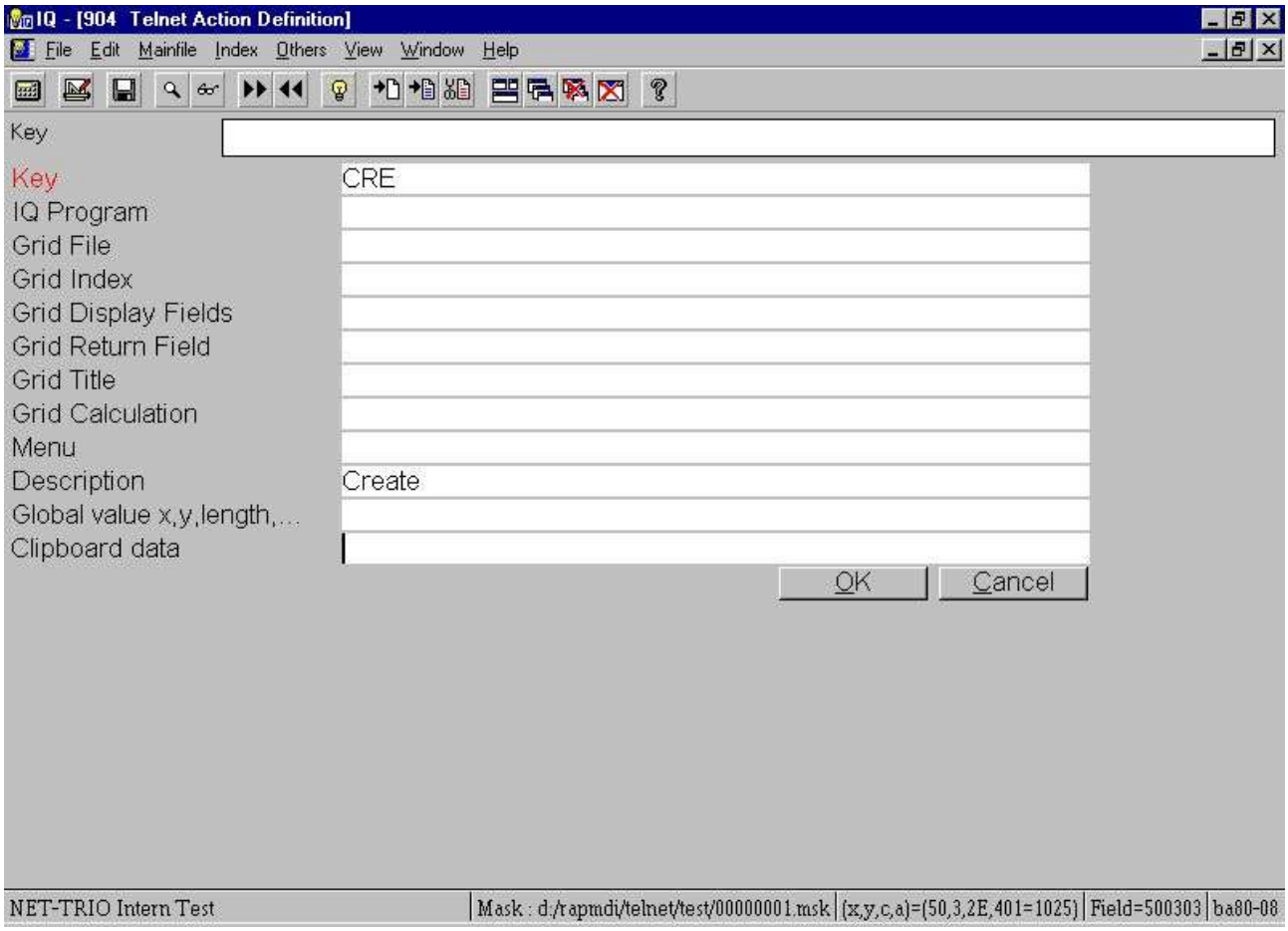
- Press CTRL and click left mouse button

Now the DATAMASTER application 904 is started and a you may now enter all the 3-letter command in the field **Menu**.



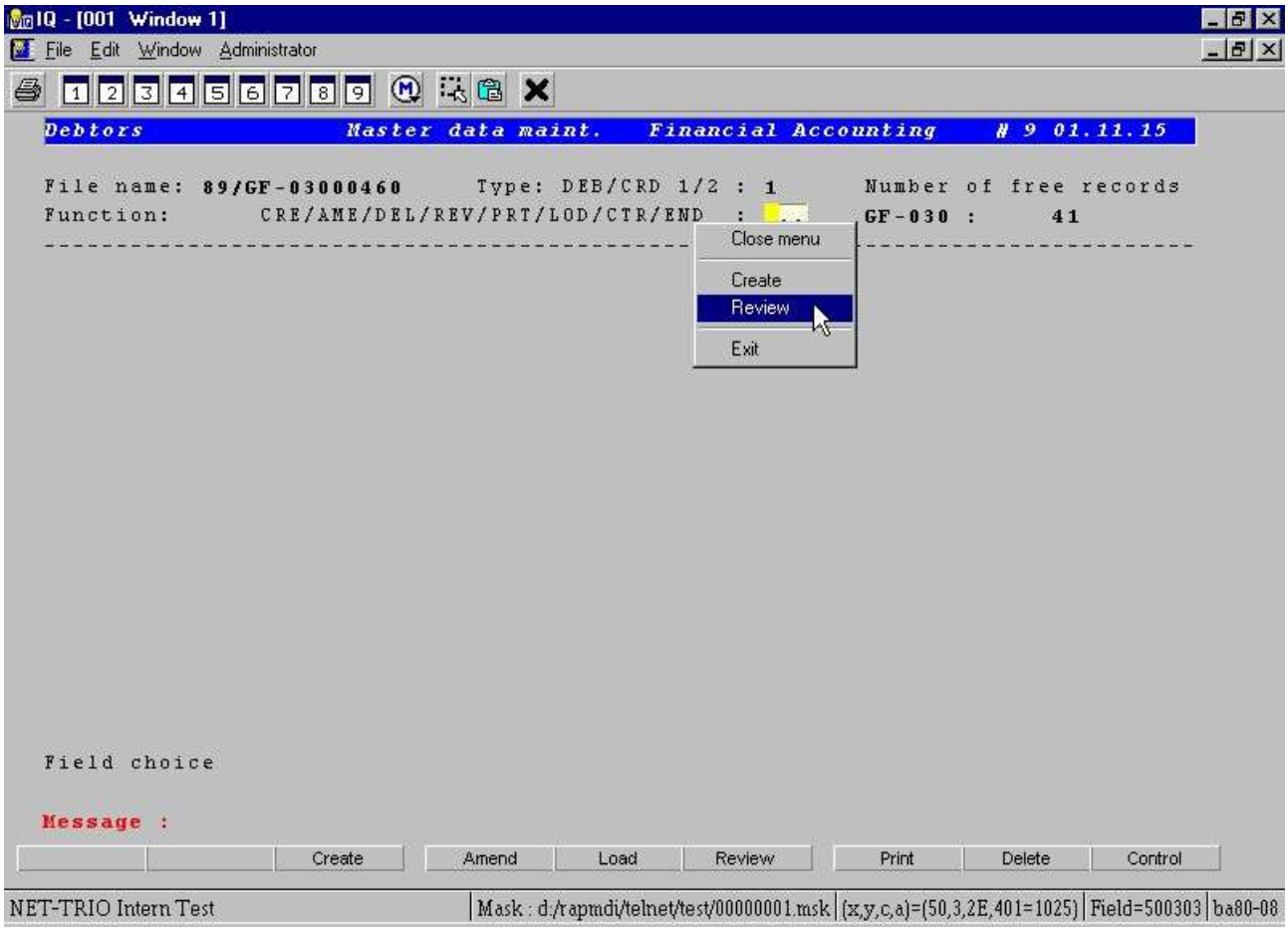
8. Step 5.1 - Defining a pop-up menu for 3-letter commands

Because you want the meaningful description for each 3-letter command they must be created in the action table also. Do this by creating a separate record for each command where the key is the 3-letter command, e.g.



9. Step 5.2 - Creating a record for a 3-letter command

When close the DATAMASTER application you may now click on the field with the left mouse button which gives you the following pop-up menu:



10. Step 5.3 - The final pop-up menu

If you for example select **Review** the pop-up menu will automatically forward the command **REV** to the application. This means that the user may still just enter the 3-letter command, but may also click in the field and use the new pop-up menu.
Before getting to the next enhancement enter **ALL** in *Field choice*.

3.6. Defining a lookup field

3.6.1. Background change equals new mask

The application have now extended its background fields to include a number of debtor fields, starting with an input of a debtor number. Because the content of the background have changed you need to get a unique mask number one again.

- Clicking on the first column on the first line, e.g. the topmost left position of the application window.

and follow the steps described in this chapter under *Unique mask number*.

Now this field enable the user to enter a debtor number which in many situations is hard to remember to a user of the system. Therefore many applications have an additional search function to enable the user to lookup the debtor according to name, address or search field.

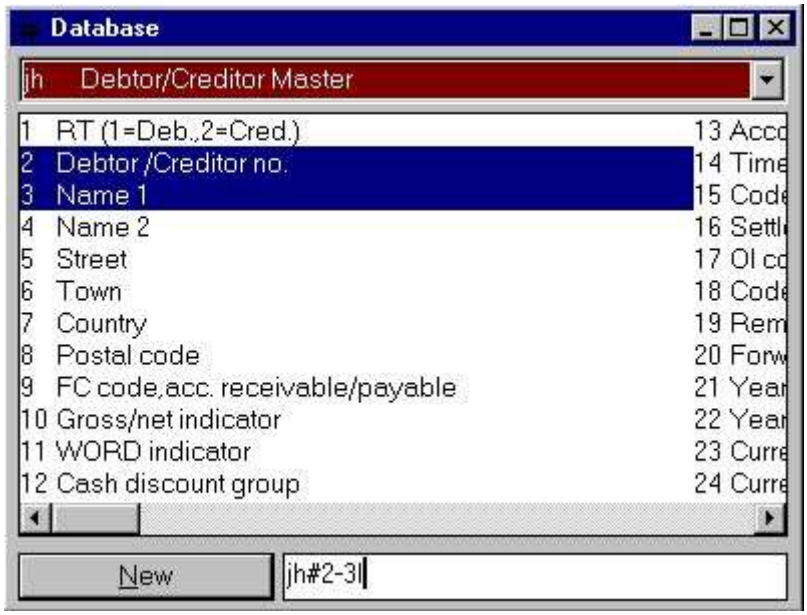
However, many applications have several application masks where the search functionality is not present due to limited space or limitation within the application size.

Also, it might be that the user needs to look up information in an completely different system using another database and is therefore required to shift to another window or even machine, find the information, write it down, and then enter the debtor number.

Here comes the power of the SW-Tools TRIO package. Because TRIO is database independent and is able to access multiple databases at the same time it is able to lookup any information from any system and hereby enhance the character based application with this functionality.

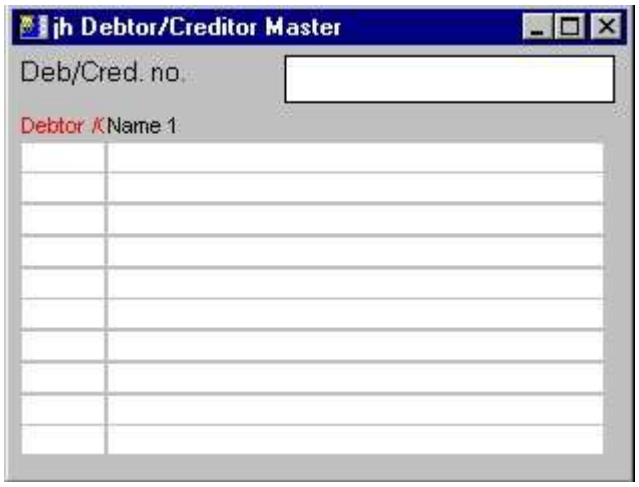
3.6.2. Defining the IQ query on debtor

Now we will define a lookup on the debtor using an IQ query with direct access to the COMET debtor master file.
First defining the query



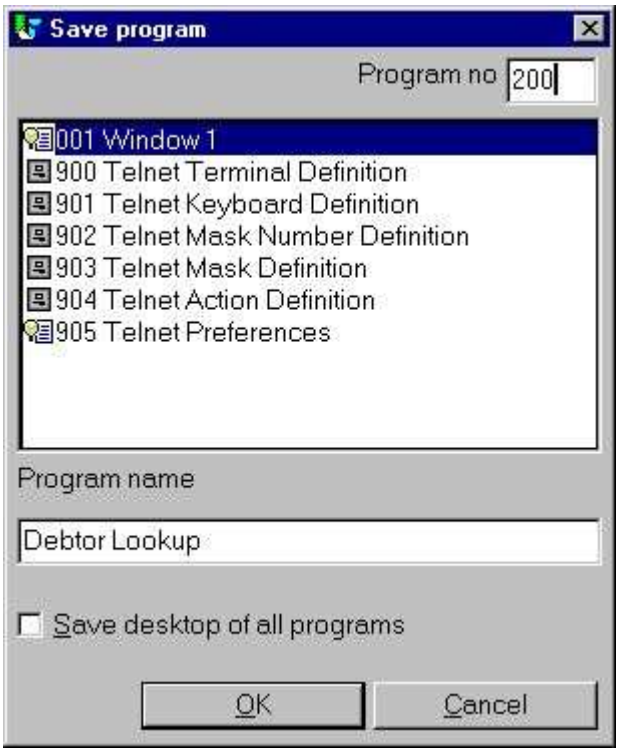
11. Step 6.1 - Defining an IQ query

will result in a query window like this:



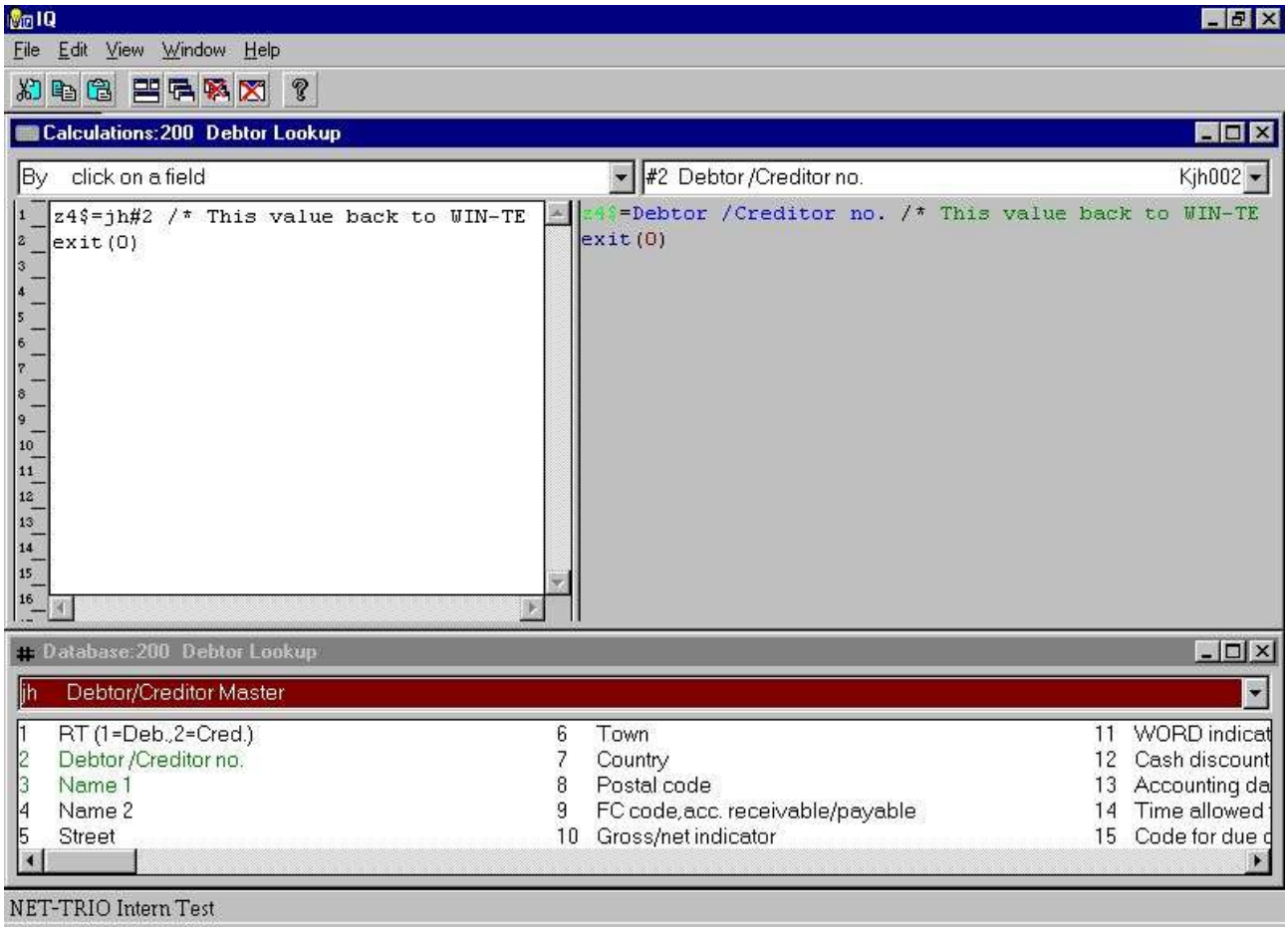
12. Step 6.2 - The query window

which is maximized (optional to do so) and saved as program number 200.



13. Step 6.3 - Saving the query

In order for the IQ query to deliver any value back to WIN-TE is must be stored in the global IQ variabled named z4$.



14. Step 6.4 - Calculations to return value to WIN-TE

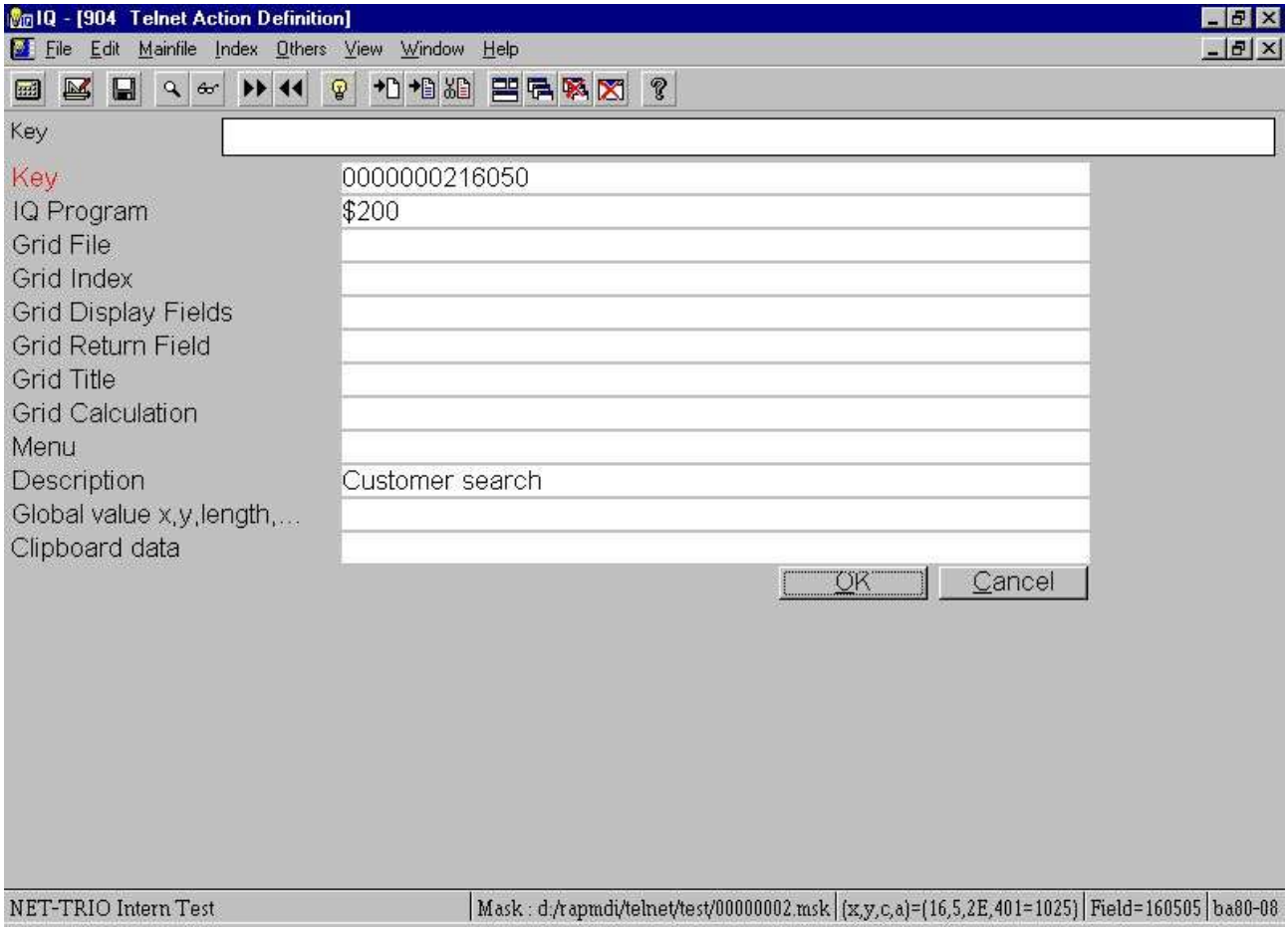
So this query enables the user to free text search for any customer and return the debtor number to WIN-TE.

3.6.3. Defining the action on the debtor field

Now you want to call this IQ query 200 when the user clicks the debtor field. So you need to define a new action for this input field.

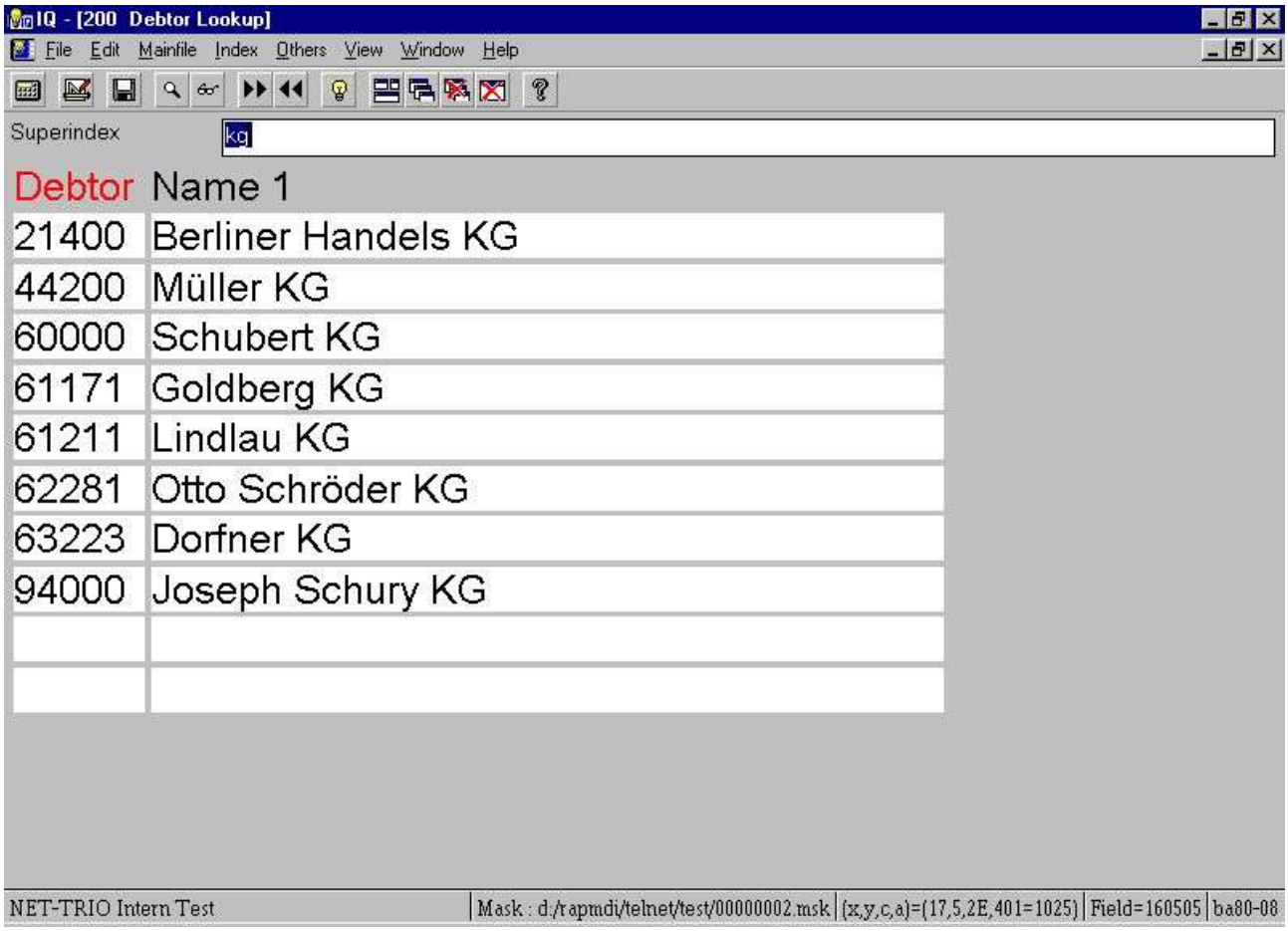
- Press CTRL and click left mouse button

and enter the following data:



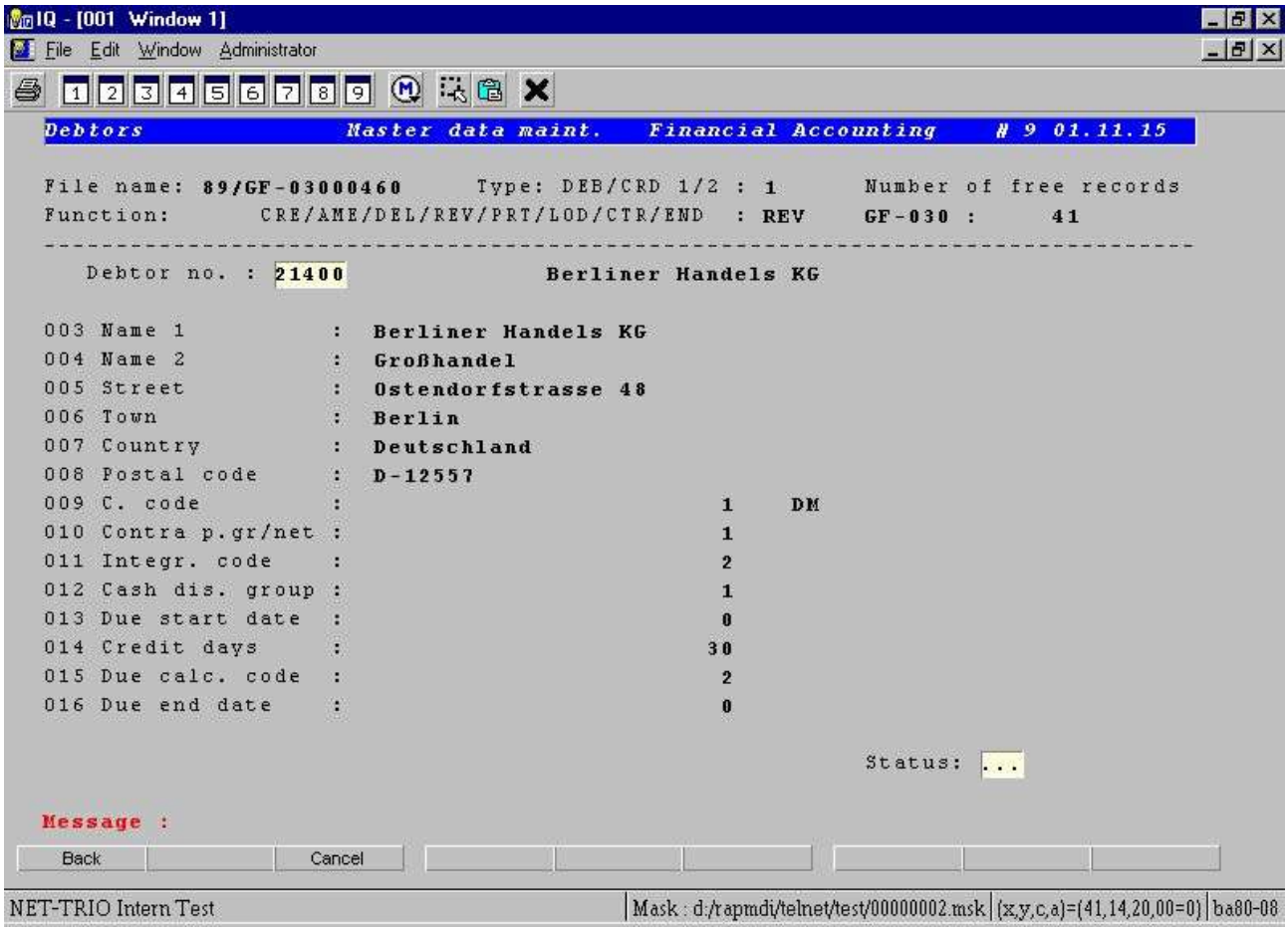
15. Step 6.5 - Action for click on debtor number field

close the DATAMASTER program and left click in the debtor field. You will now jump directly into the defined IQ query and is able to search for a debtor.



16. Step 6.6 - Finding and selecting the debtor

When you have found the debtor you may click on the debtor number and it will be returned to WIN-TE.



17. Step 6.7 - Application displays debtor data

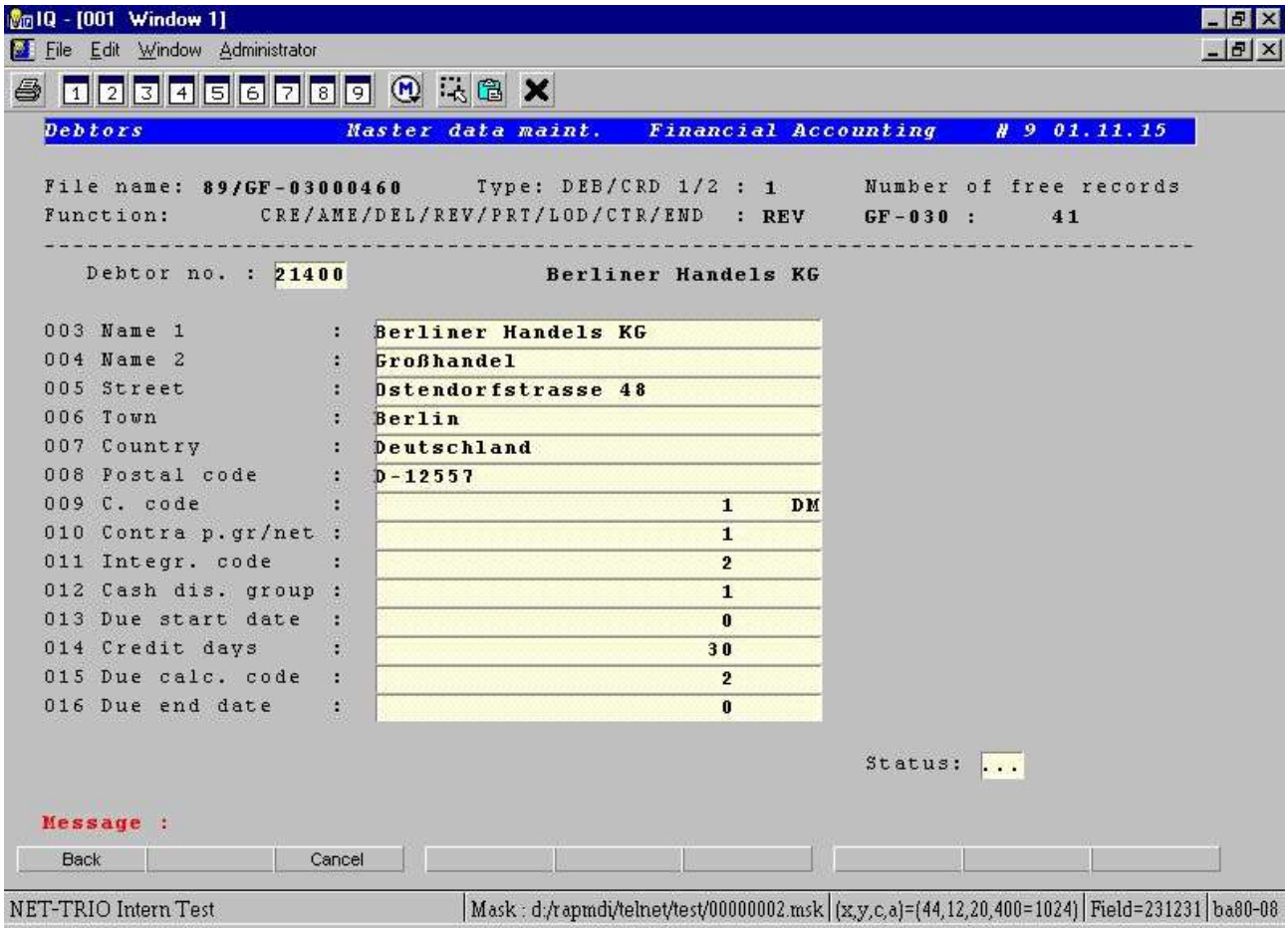
As with the pop-up menu enhancement this enhancement also enable the user to work with the system as before but is now able to work with the IQ query if clicking in the field.

3.7. Defining input fields manually

Window applications present input fields in a white background with a 3D effect on the input box. To accomplish this in the character base application you need to define a field within the mask as an input field.
In the example several field could be defined as input fields. Each field may be defined with the following procedure:

- Press SHIFT and click left mouse button at the position where the field start
- Click the left mouse button at the position where the field ends

Immediately the field is defined and displayed according to the input field setting in the preferences.



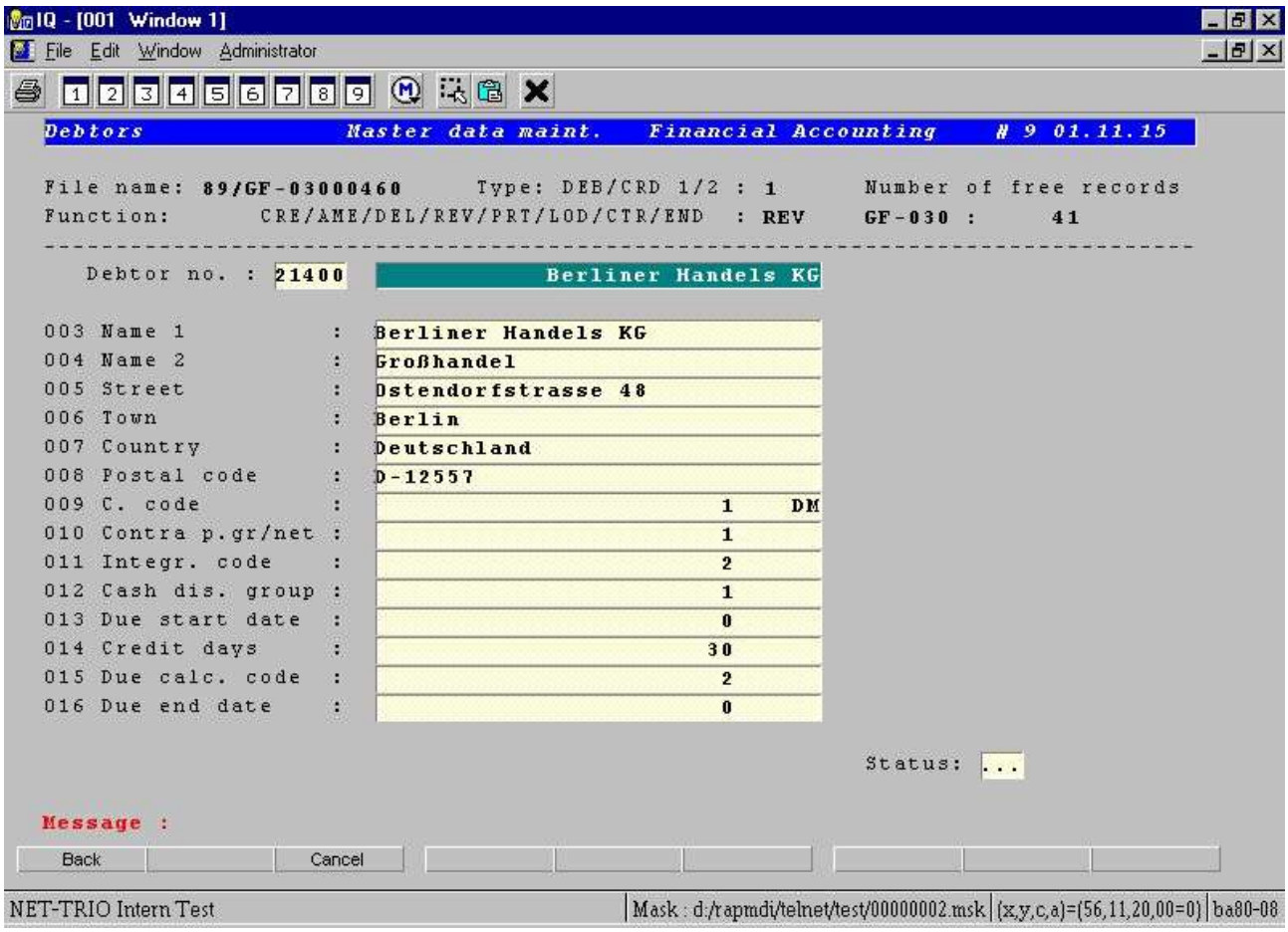
18. Step 7 - Defining input fields manually

3.8. Defining display fields manually

The display fields is the same as the input fields, but the idea is to be able to have separate color settings for the two and not allowing the user to click in display fields.
A display field may be defined with the following procedure:

- **Press SHIFT and click left mouse button at the position where the field start**
- **Press SHIFT and click the left mouse button at the position where the field ends**

Immediately the field is defined and displayed according to the display field setting in the preferences.



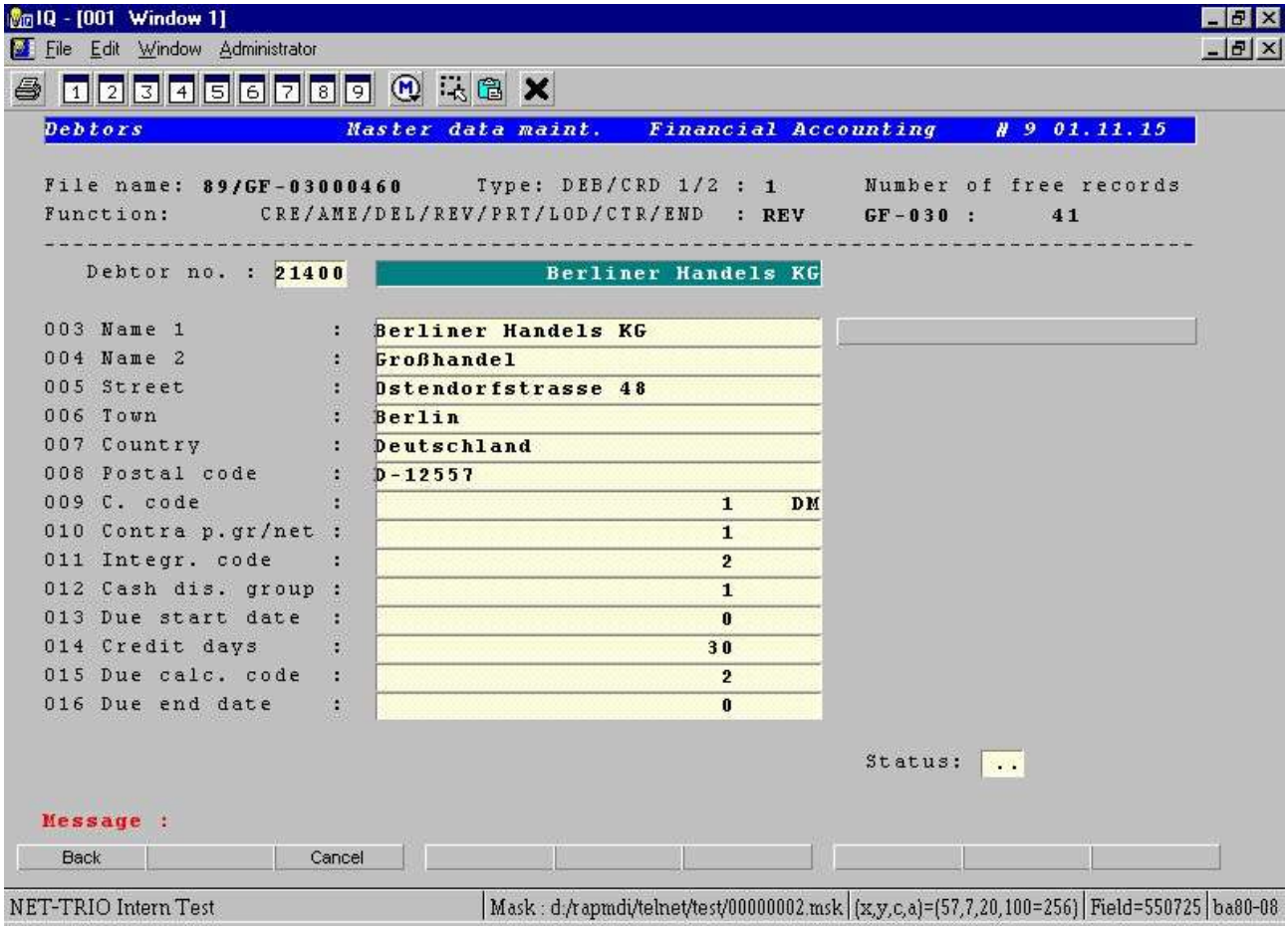
19. Step 8 - Defining display fields manually

3.9. Defining button fields

A button field is a completely new type of field that only exist in the enhancement within WIN-TE. It enable you to define areas on the screen where the user may then click to perform any kind of action not known to the current character based application. It may be useful in order to display pictures, play video or sound etc.
A button field may be defined with the following procedure:

- Press SHIFT and click left mouse button at the position where the field start
- Press CTRL and click the left mouse button at the position where the field ends

Immediately the field is defined and displayed according to the button field setting in the preferences.



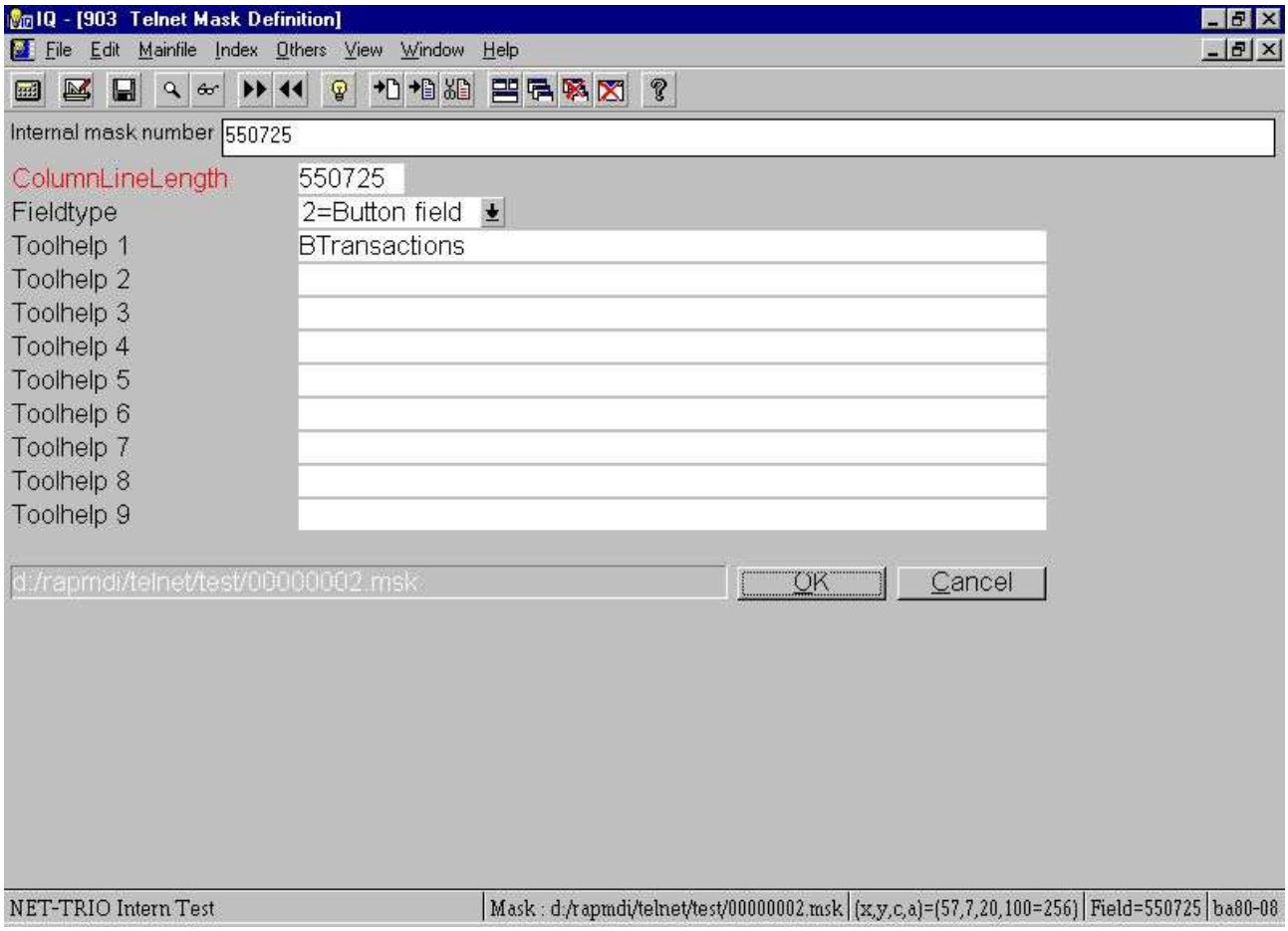
20. Step 9.1 - Defining a button field

However the button does not have a text which may be defined by using the following procedure:

- Press ALT and click left mouse button at button field

This will execute DATAMASTER application 903 where the mask definition is maintained. Here you may enter the text of the button in the first of the 9 tooltip fields. The button text MUST be entered with a leading capital B, e.g.

BTransactions

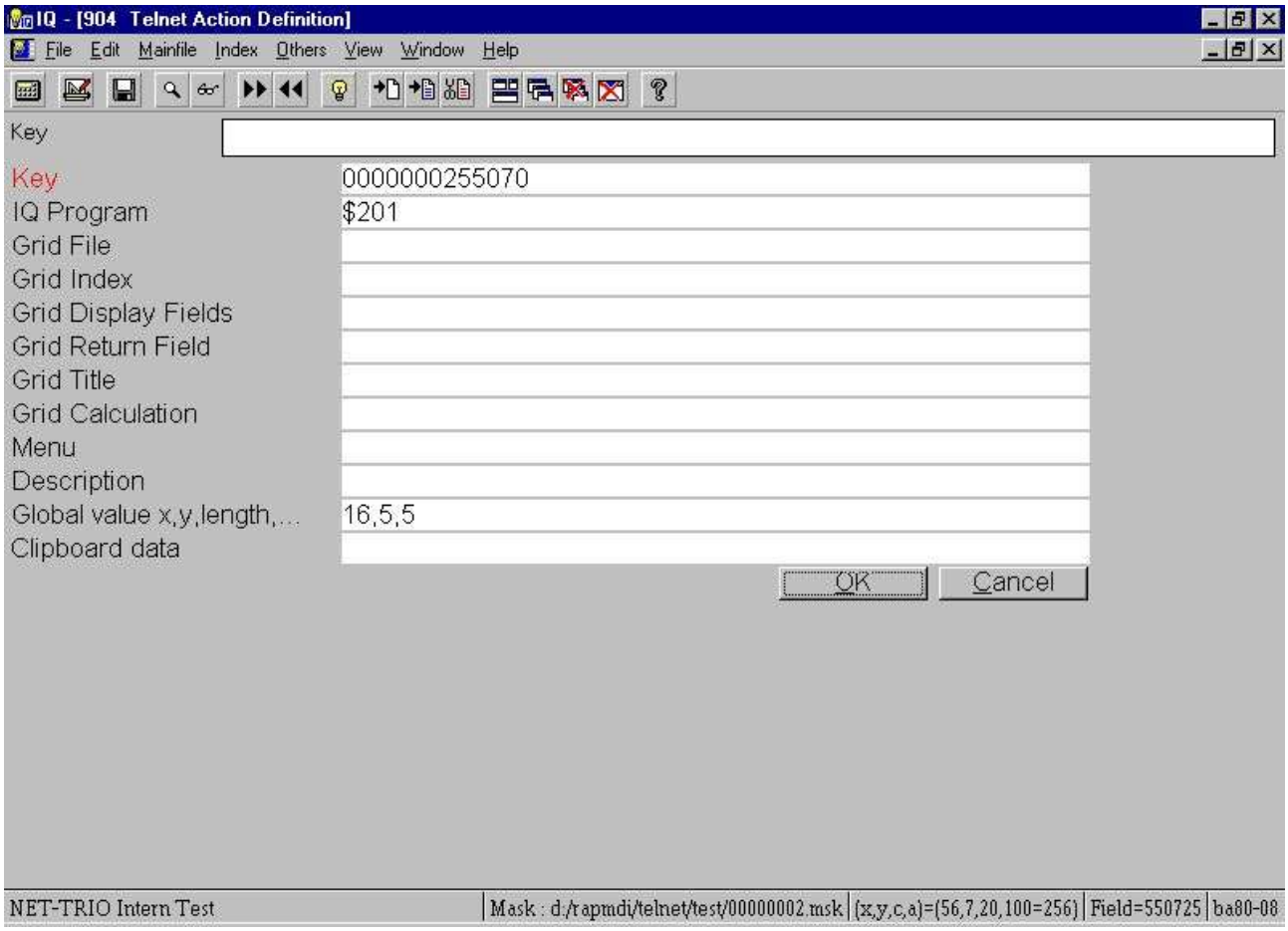


21. Step 9.2 - Defining a button field text

Now you want to call the IQ query showing debtor transactions, (lets say its IQ program 201) when the user clicks the debtor field. So you need to define a new action for this button field.

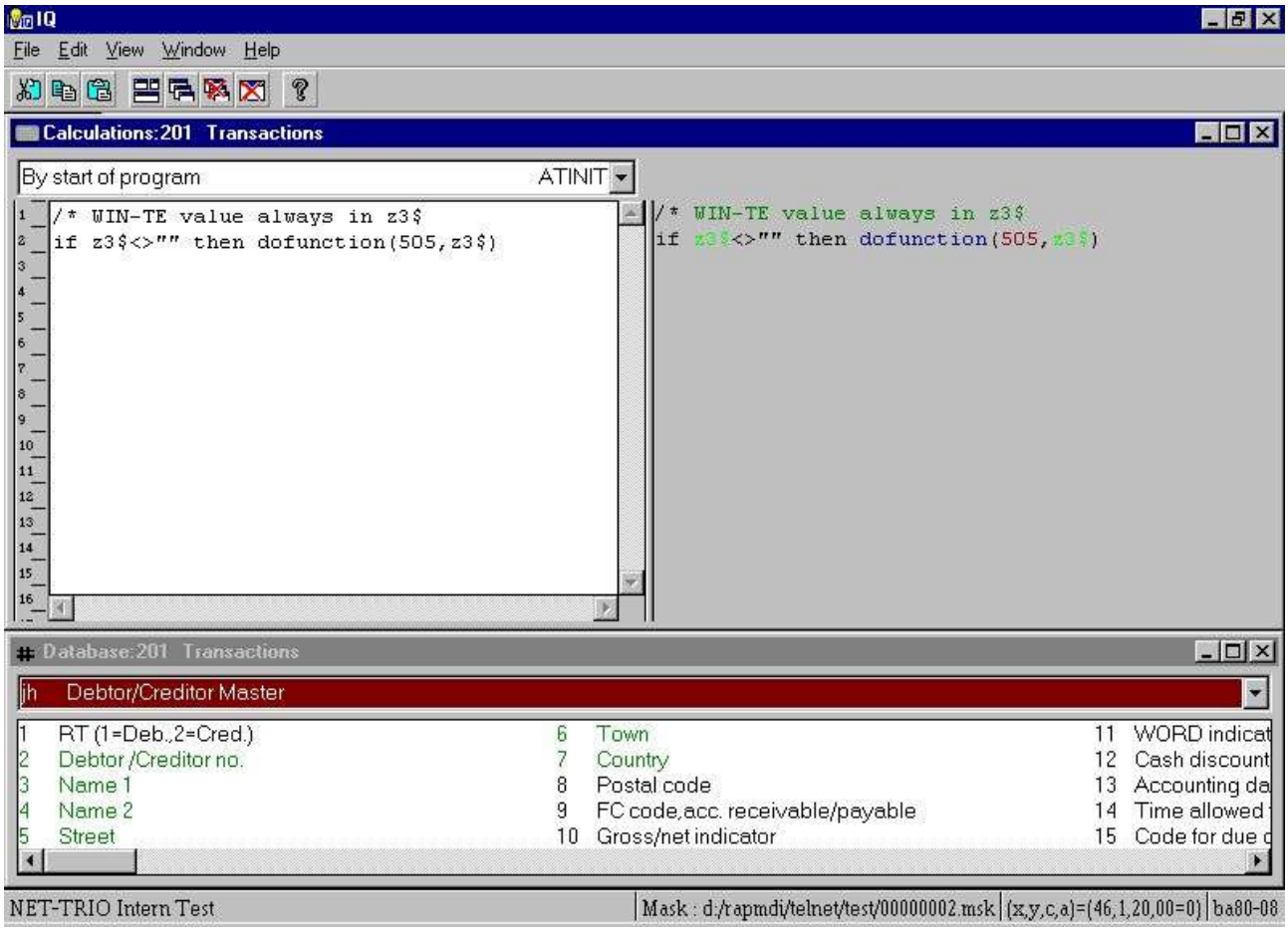
- Press CTRL and click left mouse button

and enter the following data:



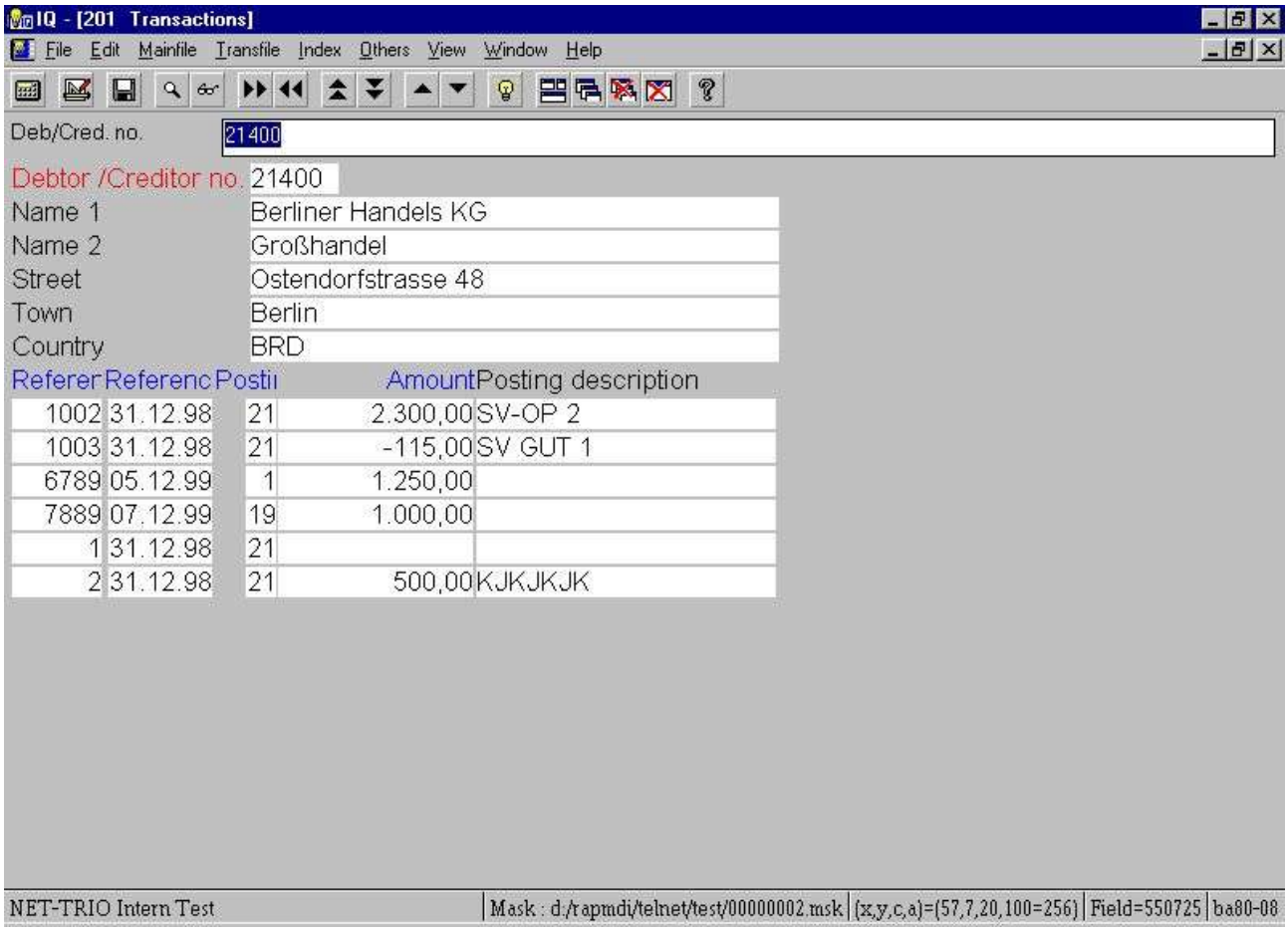
22. Step 9.3 - Action for click on button

Now the reason for the global value is that you want to pass the debtor number from the character based application to IQ in oeder to lookup the customers transactions directly without having to enter the customer number once again. The field **Global value x,y,length** is the screen coordinates for the location of the debtor number and the length is how many characters it need to include when copying.
The IQ program then need the following calculations in order to interact correctly with WIN-TE:



23. Step 9.4 - IQ Calculations required to retrieve values from WIN-TE

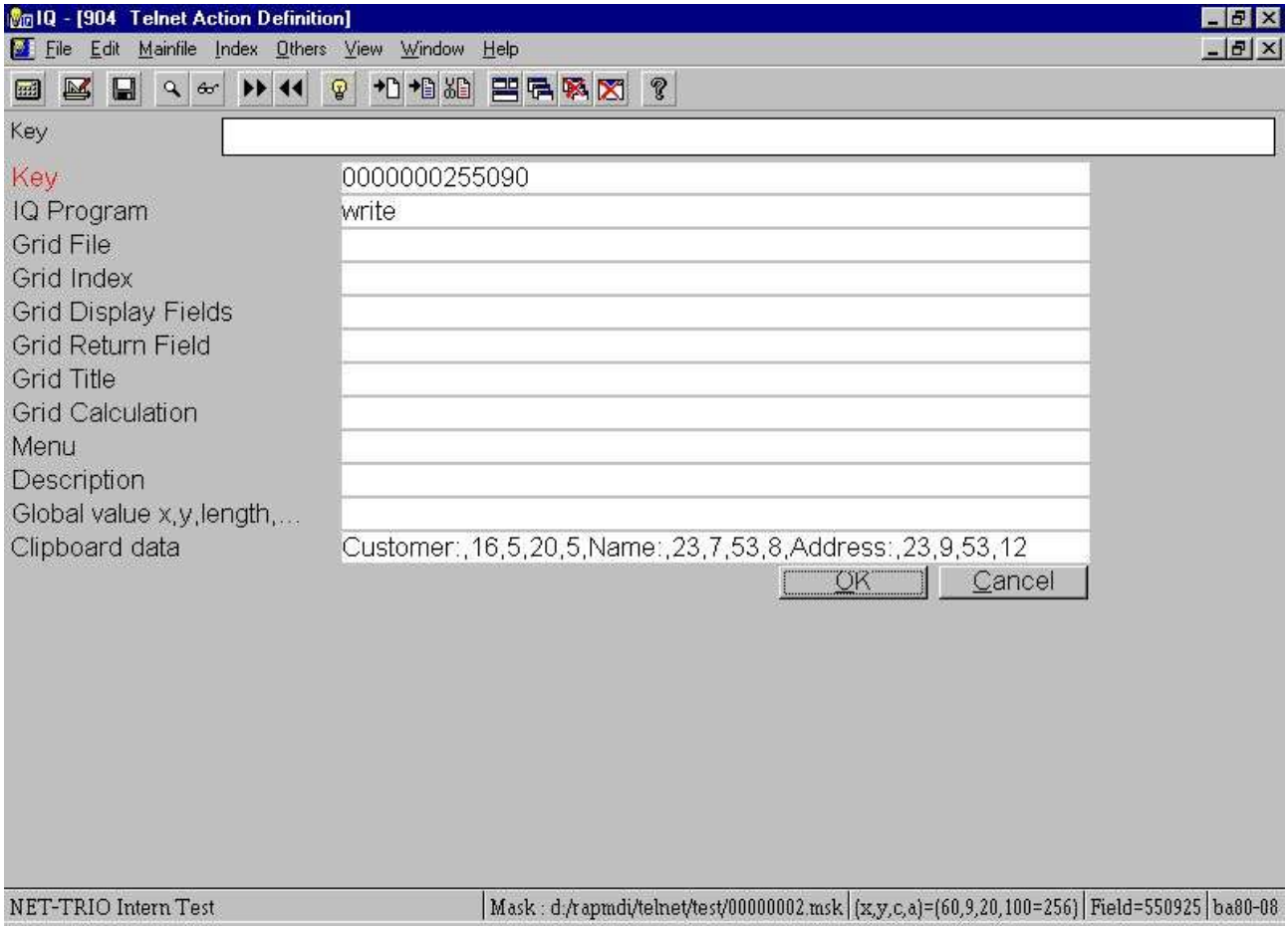
Now this is what happens when you click on the button:



24. Step 9.5 - Application enhanced with button

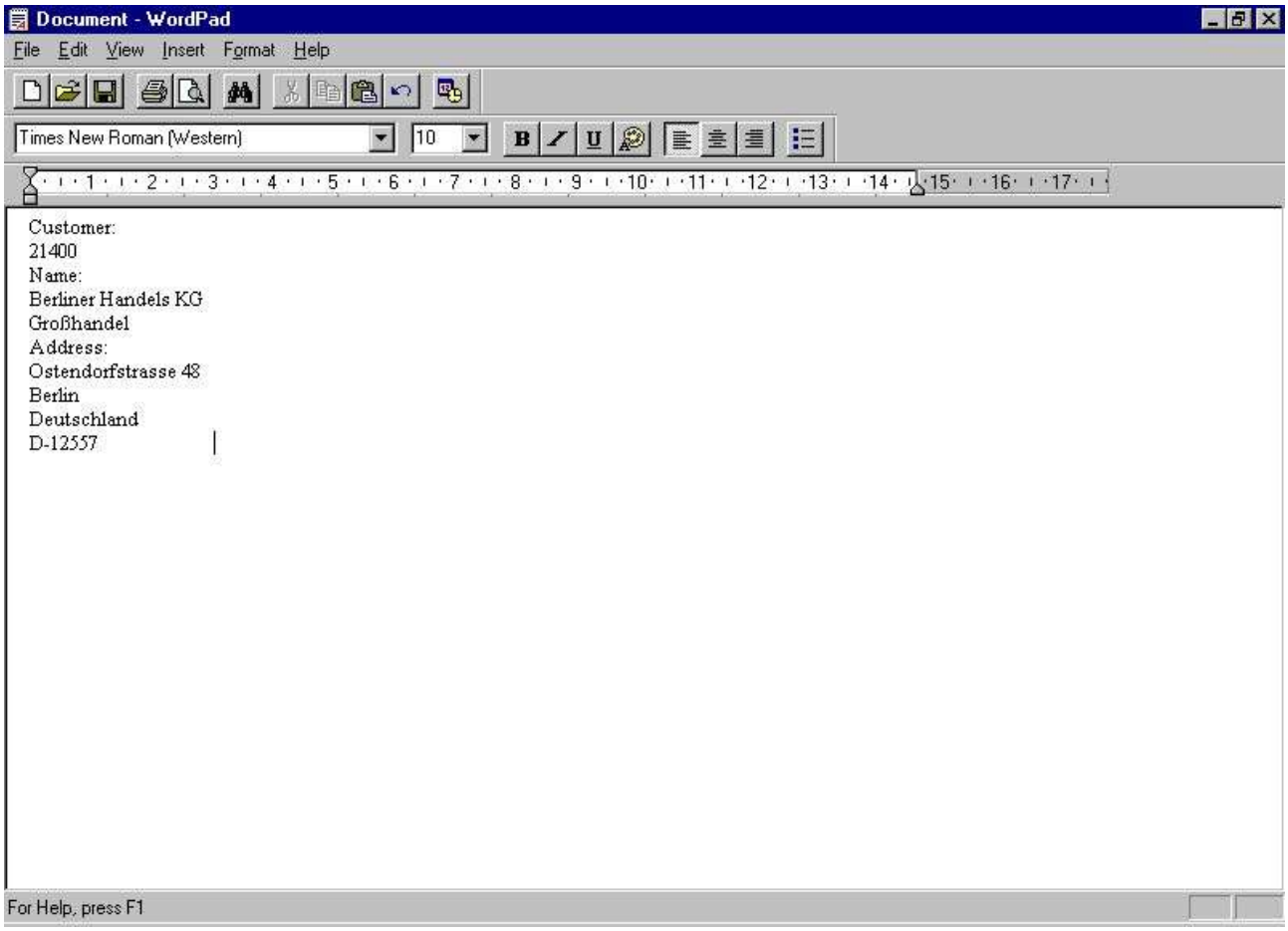
3.9.1. Copy application text and call window application

Now assume you want to write a letter to this debtor it would be nice to use a word processing application, but at the same time have the debtor number, name and address information at hand.
You could accomplish it in another emulator by copying the text area that contains the debtor data and then run the word processing application.
This can be done much easier within WIN-TE by defining yet another button just like before, but instead of getting the debtor number for use in IQ we copy multiple areas of the application screen into the Windows System clipboard.



25. Step 9.6 - Action for click on button with copy of text

If you click on the button write will be started and all you need to do now is to press CTRL+V to get the following result:

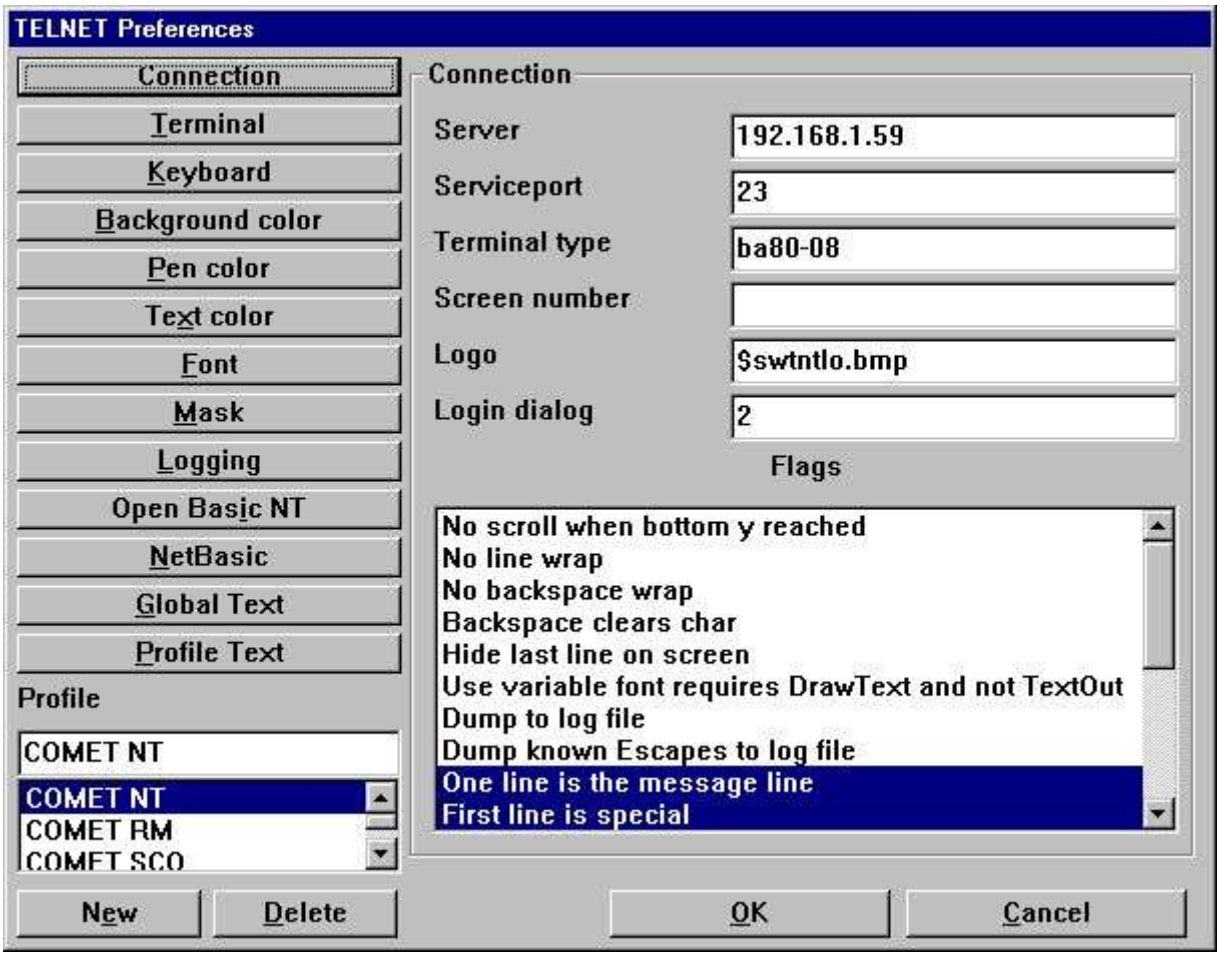


26. Step 9.7 - Write started and debtor information inserted

4. Preferences

4.1. The preference dialog

You may find the preference dialog in the IQ program 905, but also within the emulation window menu **Administrator** - **Preferences...**.



27. Preferences

All preferences is stored in the file **SWTNT.ENG**.

4.2. Profiles

The preferences for the TELNET object may be stored under individual profiles. This enables you to define a profile for different UNIX connections, e.g. one for SCO UNIX and one for LINUX. Each profile is identified by a unique name.

4.2.1. How to create a new profile

The creation of a profile is done by

- selecting a profile from the list
- entering a new profile name
- selecting New

The new profile may then be modified.

4.2.2. How to delete a profile

Delete a profile by selecting the profile from the list and then select **Delete**.

4.3. Connection

4.3.1. Server

This field may contain the IP-address or name of the server. If using a name it should be listed in the hosts file, e.g. (c:\windows\hosts).

4.3.2. Service port

This field contains the port number required to access the TELNET service on the host computer. Please refer to the /etc/services file on UNIX for the service 'telnet'. Normally the port number is 23.

4.3.3. Terminal type

This field contains the actual name of the terminal type used. The TELNET object forwards this when connecting to the telnet service.

4.3.4. Screen number

This is an optional field. You may use this to represent a logical screen number. The number may be retrieved using the function TelnetGetNum().

4.3.5. Logo

The field contains the full path of a Windows Bitmap File. This logo will be displayed in the login dialog. The size of the bitmap is 196 wide and 76 tall in dialog units (Normally x dialog units equls x*2 pixels and y dialog units equals y*2 pixels).
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().

4.3.6. Login dialog

This field represents a number from 1 to 4. Each number equals a predefined dialog:

1 - Dialog contains Login name, Password
2 - Dialog contains Server, Profile, Login name, Password
3 - Dialog contains Password
4 - Dialog contains Server, Profile, Password
5 - Dialog contains Server, Password

Additional all predefined dialogs contains a Logo area and a button for Login and Exit.

4.3.7. Flags

This field enables you to select one or more options which controls the way the TELNET object work.

Flag	Description
1	No scroll when bottom y reached
2	No line wrap
4	No backspace wrap
8	Backspace clears char
16	Hide last line on screen
32	Use variable font requires DrawText and not TextOut
64	Dump to swtnt.log
128	Dump resolved escape sequences to log file
256	One line is the message line
512	First line is special
1024	Last line is special
2048	No 3D effect if single space
4096	Hide emulator screen
8192	BASIC line 0 is hidden
16384	Dump unknown Commands and Escapes to log file
32768	Wif is on
65536	Auto mask fields
131072	No dot if input field
262144	Display column/line grid
524288	Show information in statusbar

Flag 1:
When the emulator reaches the last line on screen it normally wraps the screen content, e.g. it deletes the first screen line and scrools all other lines one up leaving the last screen line blank. If this flag is set it will not wrap the screen.
Flag 2:
When the emulator reaches the last column on line it normally advances to the next line, e.g. it jumps to the first column on the next line. If this flag is set it will not wrap by line end.
Flag 4:
By default the emulator will wrap to the previous line if backspace is used on a line in the first column. If this flag is set it will not wrap to the previous line.
Flag 8:
If this flag is set the backspace will clear the character, e.g. set it to a space, if not the emulator will simply go one column to the left.
Flag 16:
This field may be used to hide the last line. Let us assume that your application only uses 24 lines in total, but the terminal type is set to 25 lines. By setting this flag you will only get 24 lines displayed.
Flag 32:
If you use any variable width fonts in the preferences make sure to set this flag. If you do not the content displayed may not be correctly setup in columns.
Flag 64:
If this flag is set the entire flow between the telnet service and the TELNET object will be stored in the swtnt.log file. The filename swtnt.log may be modified to another filename in the preferences. By default it is stored in the same directory as the SWT00832.DLL.
Flag 128:

If this flag is set any resolved escape sequence is dumped to the log file. This is useful when defining new terminal definitions.

Flag 256:

If this flag is set the TELNET object will assume that one of the lines on screen is a message line. Then it is possible to set the background and foreground color for the message line. The actual line number may be set elsewhere in the preferences.

Flag 512:

This flag marks the first screen line as being special. It is used to mark the first line as a header line at all times. Then it is possible to set the background and foreground color for the line.

Flag 1024:

This flag marks the last screen line as being special. It is useful when an application for example uses the last line for SOFTKEYS. When the flag is set the emulator will inform the IQ application whenever changes are made on that line by sending a notification to IQ. This notification is found in the calculation section "Telnet Object" + "Last screen line changed".

Flag 2048:

The 3D effect on the screen is controlled by the fonts setup in the preferences. However, you may need to avoid 3D effect to be made if just a single space triggers it. The problem is found in BUSINESS BASIC applications that uses background and foreground to control the text and input fields. They set one background space to limit the length og a field which then causes a 3D effect to be displayed by the emulator on that one space.

Flag 4096:

This flag will hide the emulator screen which is useful if you simply want to login and control all by calculations without having any interfearence from the user.

Flag 8192:

No documentation yet!

Flag 16384:

This will dump any unknown escape sequence into the log file. This is useful when defining new terminal definitions.

Flag 32768:

Reserved for use by SW-Tools ONLY.

Flag 65536:

The automask field option is intended to be used during development only. It enables you to define the mask definition. Instead of having to edit the mask file manually you may use the keyboard in connection with the mouse to define which areas of the screen contains input, display and button fields. Please refer to the chapter regarding automask functionality.

Flag 131072:

The flag will supress dots in any field area marked as an input field within the current mask. BUSINESS BASIC applications normally displays dots where input is currently active. This flag will suppress it.

Not implimented yet!

Flag 262144:

The flag will activate a grid function which will be displayed at all times in the emulator. For every 10th column and every 5th line.

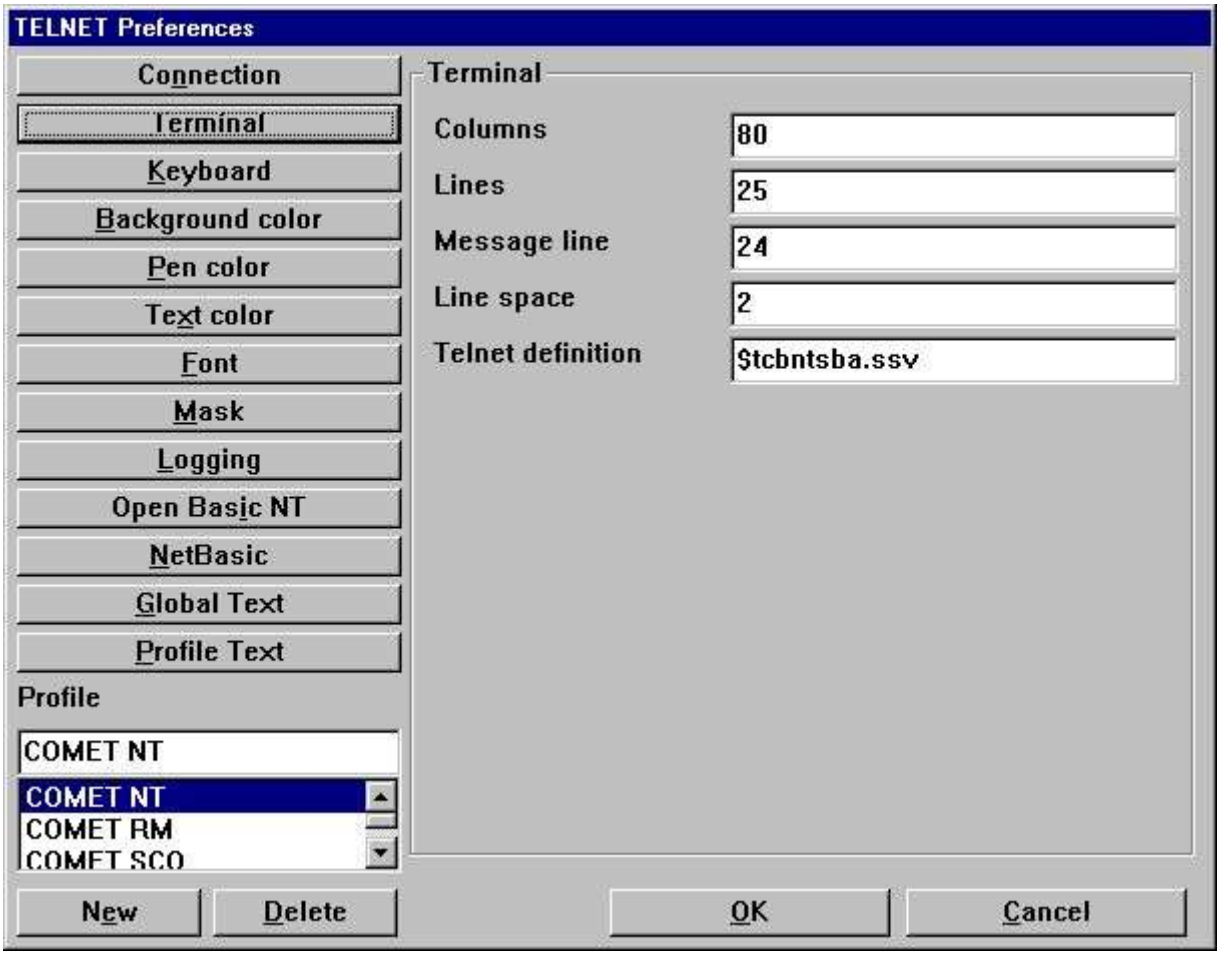
Flag 524288:

If this flag is set the following information will be displayed in the statusbar if present.

- **Terminal type, e.g. BA80-08**
- **Column,Line,Character,Attribute according to mouse position, (x,y,c,a) = (10,12,42,40=64)**
- **Mask field according to mouse position, e.g F : 260305**
- **Current mask file if any, e.g. M : c:\swtools\winte\eng\00000001.msk**

Please note that the (x,y,c,a) displays the Attribute (a) value as Hexadecimal=Decimal, e.g. 40 = 64.

4.4. Terminal



28. Preferences - Terminal

4.4.1. Columns

Represents the number of characters on one line.

4.4.2. Lines

Represents the number of lines on the screen.

4.4.3. Message line

If the flag **One line is the message line** is set this represents the actual screen line number.

4.4.4. Line space

This value is used on Input, Display and Button fields. If set to a value greater than zero there will be one pixel line of space at the botton of such a field so that the 3D effect is not merged between lines. Also the actual value represents the number of pixels the font height should decrease on such a field to stay within the box area.

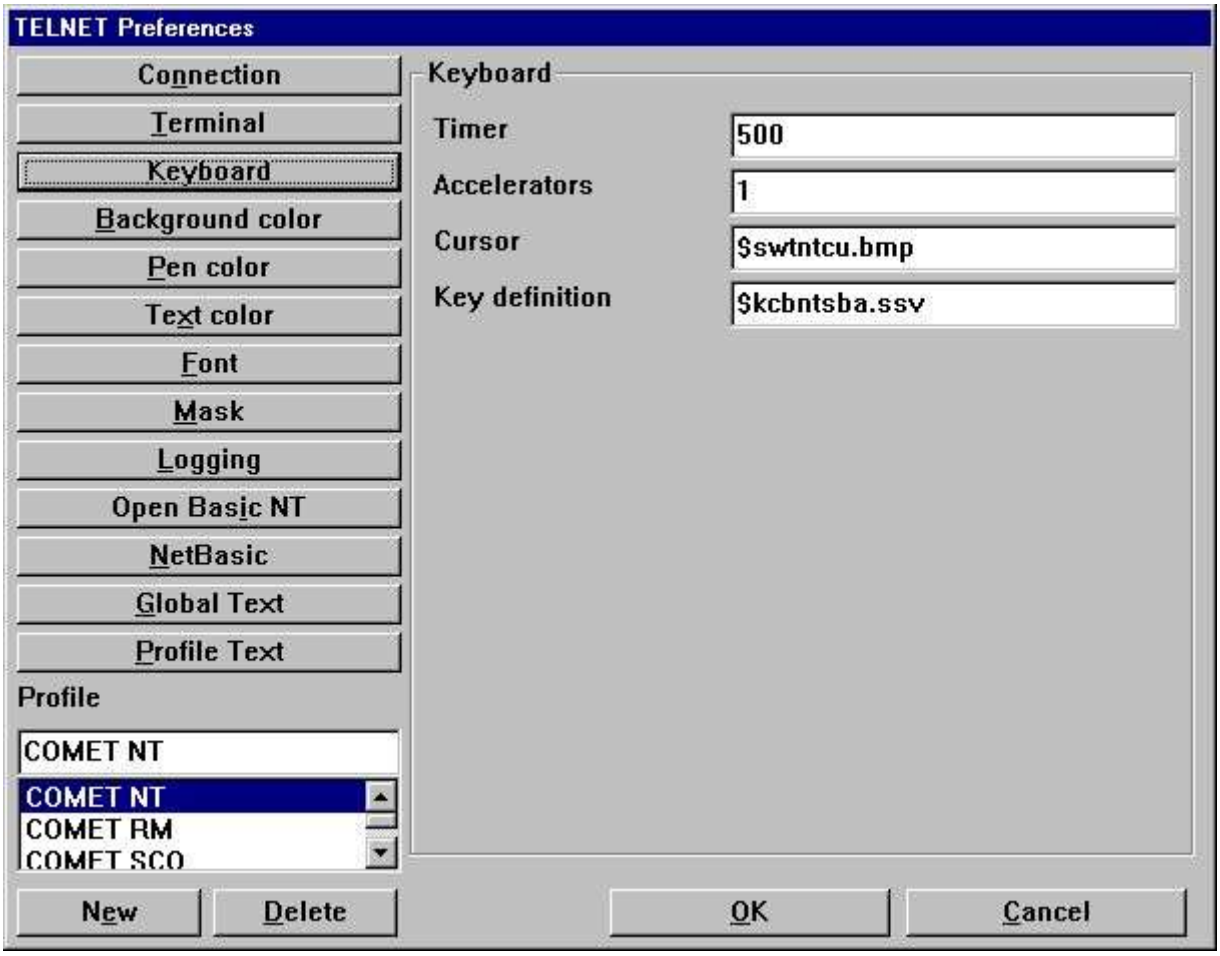
4.4.5. Telnet definition

The telnet definition file contains the definition of any ESCAPE sequence used to control the screen content. This is on UNIX systems known as the **termcap** definitions.
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().
Refer to the chapter *Terminal Definition*.

4.5. Keyboard



29. Preferences - Keyboard

4.5.1. Timer

The timer is set in miliseconds, e.g. 500 is half a second. This is the timer for the cursor blink mode and also of how often the emulator should see if screen changes have been send from the server.

4.5.2. Accelerator

This should be set to 1.

4.5.3. Cursor

The cursor field is a references to a bitmap file which contains a cursor. The bitmap size must be 12 pixels in width and 12 pixels in height. If no filename is given the cursor will be the character **_** or **|**.
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().

4.5.3.1. Animation

The cursor may also be animated. This is done simply by creating several cursor images in one bitmap file, e.g. 36 pixels in width and 12 pixels in height gives 3 animation cursors.

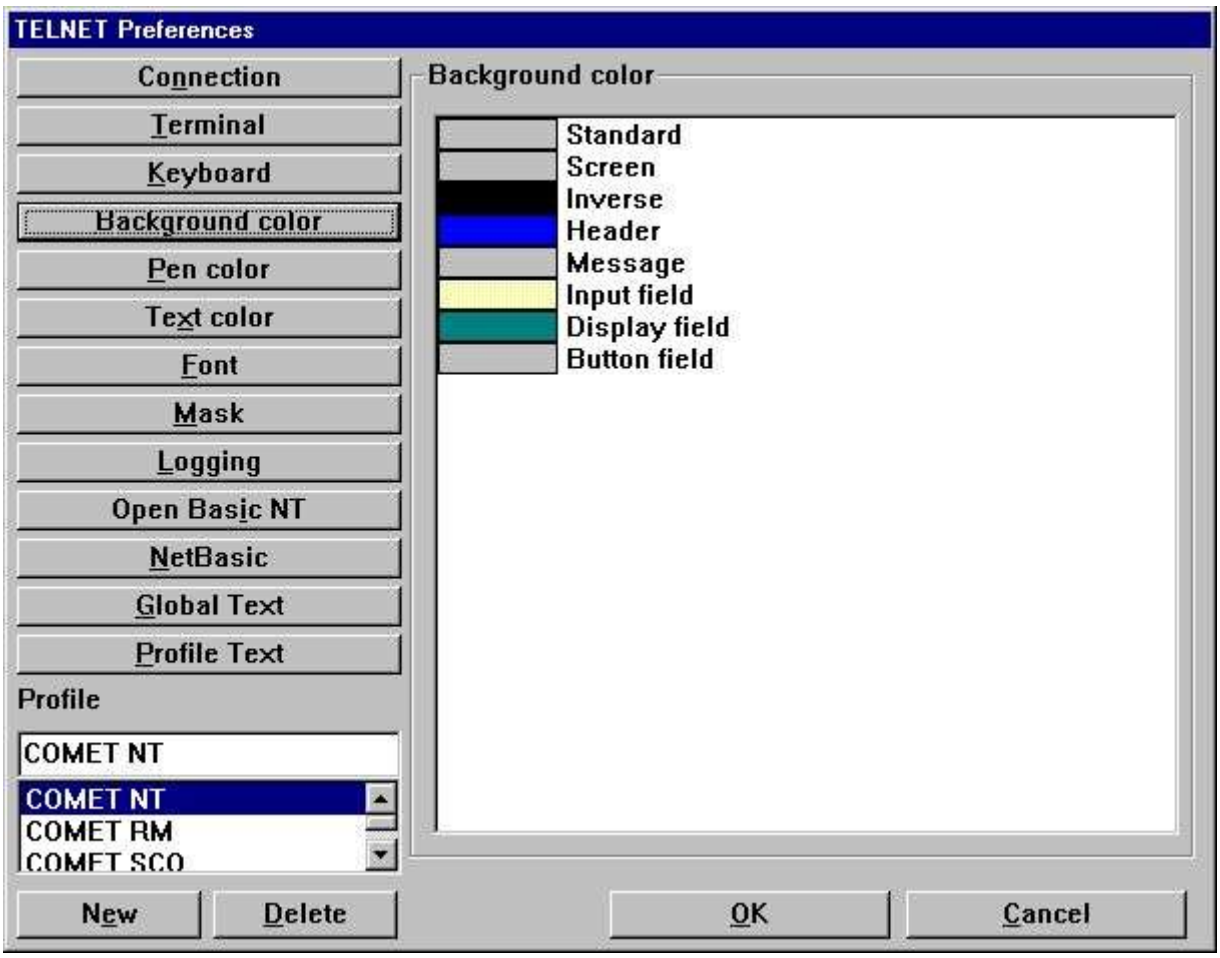
4.5.4. Key definition

The key definition file contains the definition of any ESCAPE sequence used to control the input. This is send to the server executing the application whenever a key is pressed.
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().
Refer to the chapter *Keyboard Definition*.

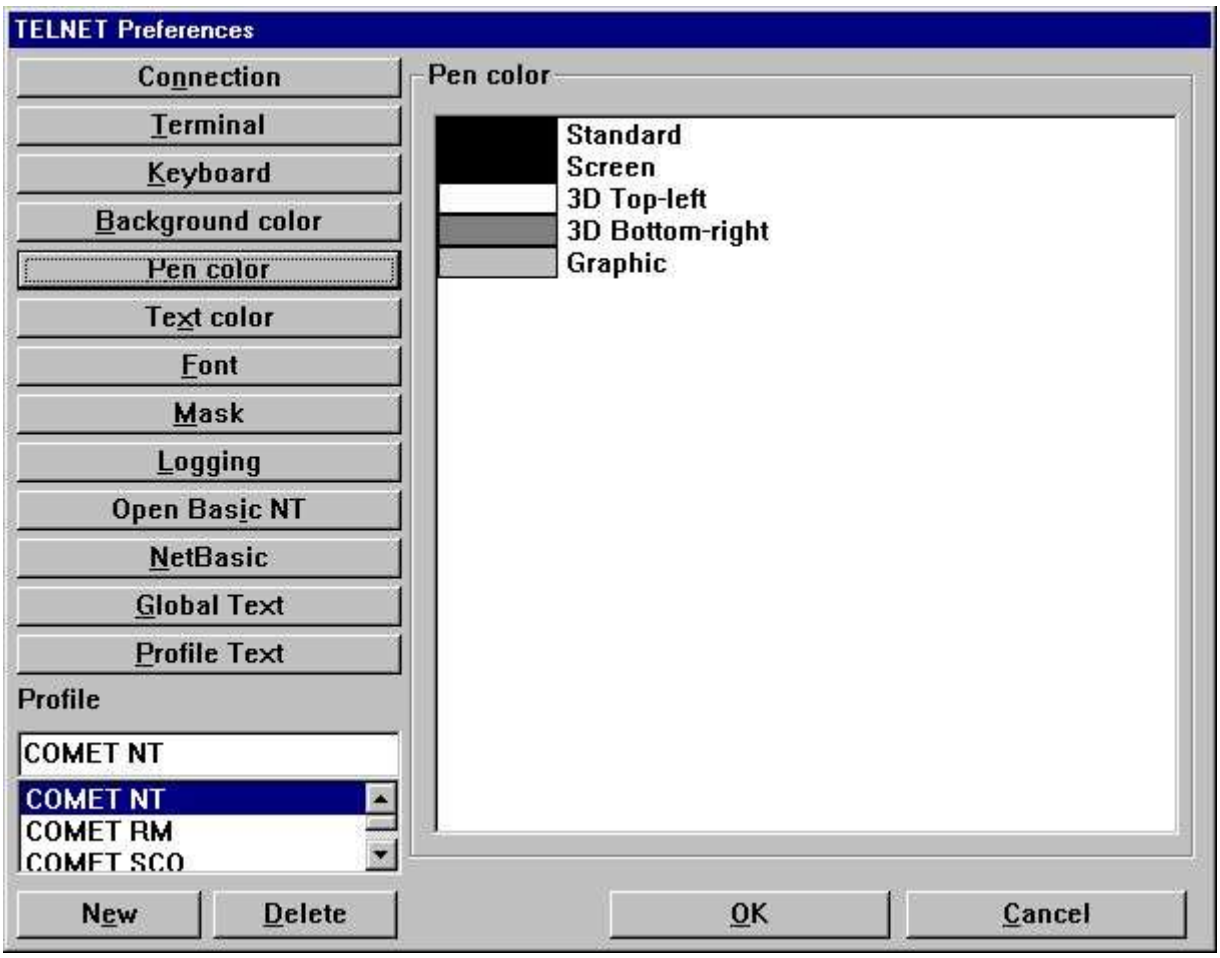
4.6. Background color



30. Preferences - Background color

The background color may be set for any of the items listed. You may change the color by double-click on the item in the list. If the color displayed is ???? in a red color is means that no color have been set for the item.

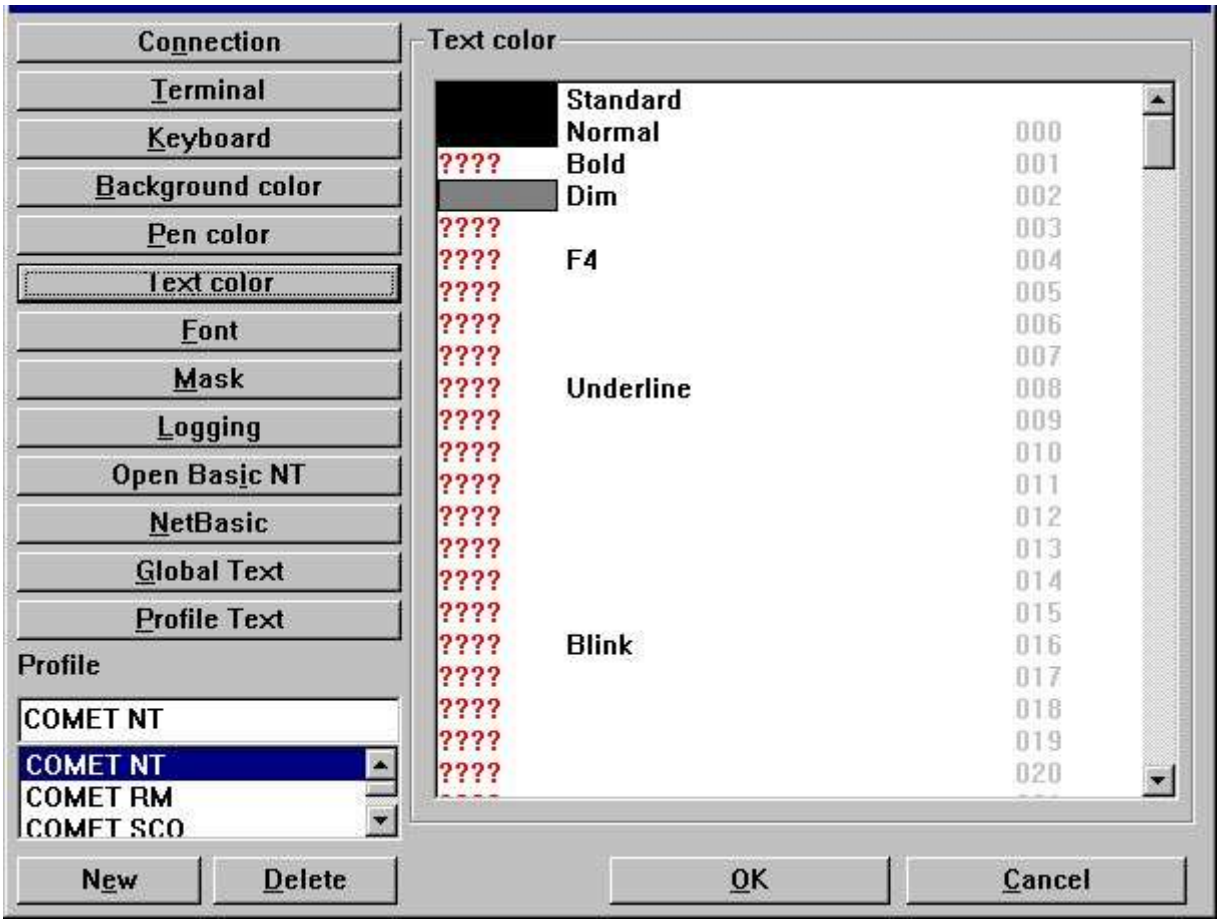
4.7. Pen color



31. Preferences - Pen color

The pen color may be set for any of the items listed. You may change the color by double-click on the item in the list. If the color displayed is ???? in a red color is means that no color have been set for the item.

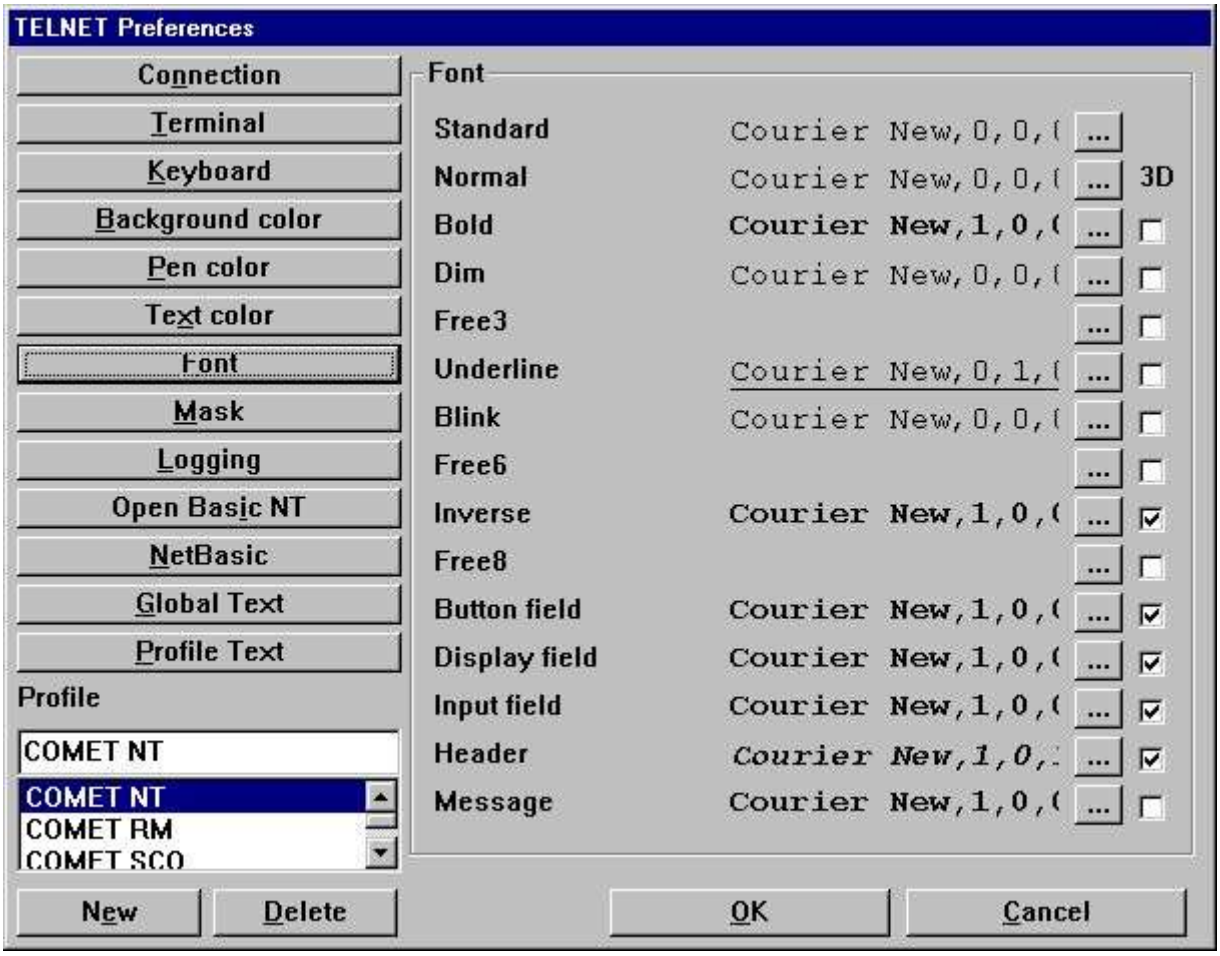
4.8. Text color



32. Preferences - Text color

The text color may be set for any of the items listed. You may change the color by double-click on the item in the list. If the color displayed is ???? in a red color is means that no color have been set for the item.
The number displayed in the right column is the actual attribute number. This number is used when setting up the mask configuration for forground and background definition.

4.9. Font



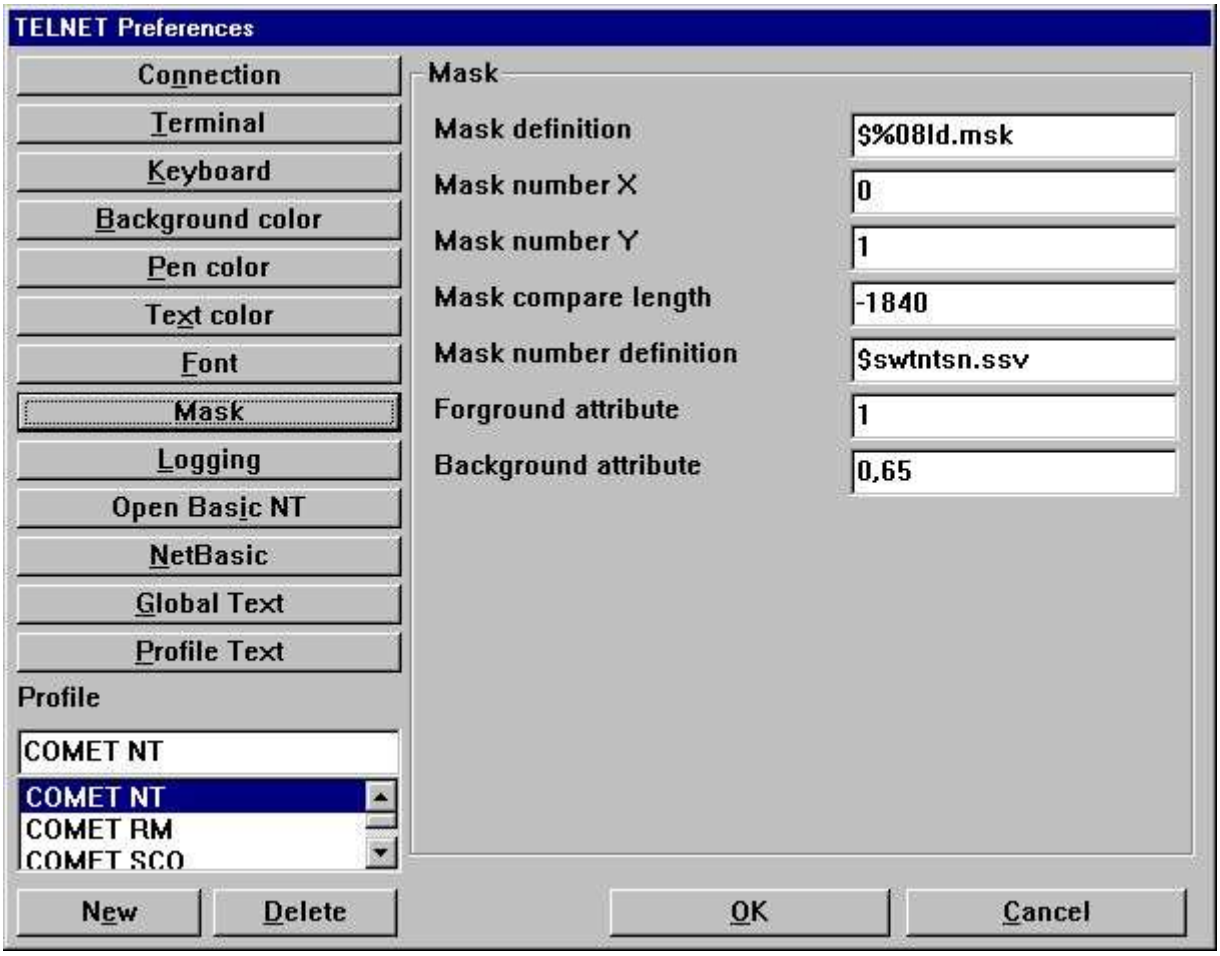
33. Preferences - Font

The font may be set for any of the items listed. You may change the font by selecting the button **...** on the item in the list.

4.9.1. 3D effect

Any of the defined font fields may also be extended with 3D effect simply by setting the checkbox for the item.

4.10. Mask



34. Preferences - Mask

4.10.1. Mask definition

The mask definition is the actual path and filename definition for any mask file used by the system, e.g.

x:\win-te\mask\%08ld.msk

will use the following file for mask number 7

x:\win-te\mask\0000007.msk

because the **%08ld** transforms the numeric mask number into a 8 digit text field with leading zeroes, where as

x:\win-te\mask\msk%ld.msk

will use the file

x:\win-te\mask\msk7.msk

If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().

4.10.2. Mask number X

This states the starting x coordinate of the screen from where the mask begins.

4.10.3. Mask number Y

This states the starting y coordinate of the screen from where the mask begins.

4.10.4. Mask compare length

This states the length, e.g. number of characters, to be used for the mask.
For example, if the entire screen background content should be used for the mask you should have

Mask number X : 0
Mask number Y : 0
Mask compare length : 2000

if the screen is 80 * 25.
If the length is given as a negative value the compare routine will only include characters a-z, A-Z and 0-9 and completely rely on the mask number definition.

4.10.5. Mask number definition

The mask number definition is used to store the screen background content in a table with a unique numeric mask number. Each time the screen background content changes it will compare the current screen with the known mask number records to find a unique mask number.
This field represents a filename of where to store the mask number definition.
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().

4.10.6. Forground definition

The forground definition is used to define which attributes that represents a forground character.
The field may include several attribute numbers separated by a comma, e.g.

1,8

Refer to the number shown in the text color list.

4.10.7. Background definition

The background definition is used to define which attributes that represents a background character.
The field may include several attribute numbers separated by a comma, e.g.

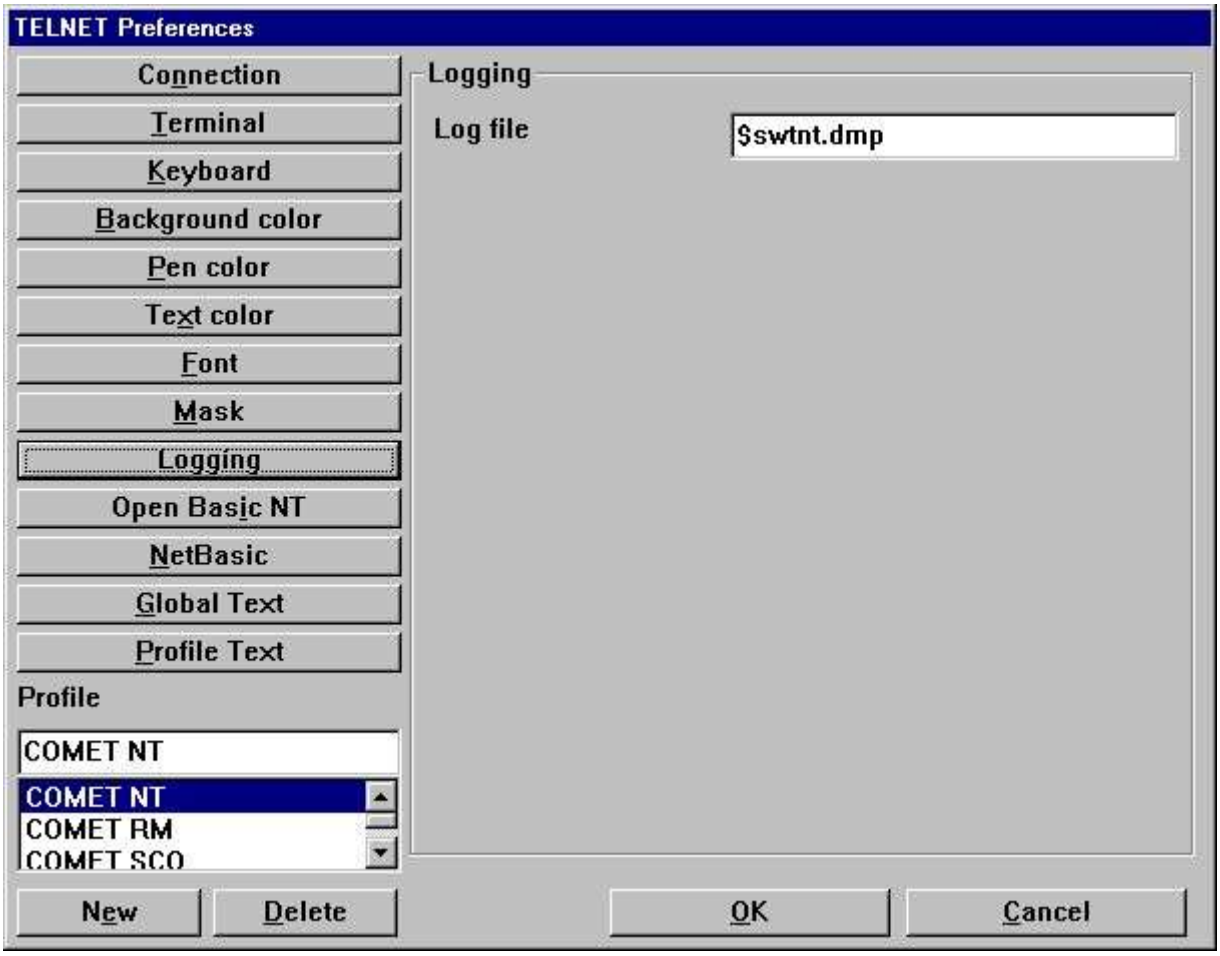
64,1

Refer to the number shown in the text color list.
For the background definition it is very important to remember that when the emulator needs to recognize a mask if will get all characters from a-z, A-Z and 0-9 where the attribute is equal to the background definition, **BUT** it will do it according to the following rules:

- Get all characters a-z, A-Z and 0-9 where attribute is 64
- If no characters found try for the next attribute number (in this case 1)

In other words, it will not merge characters from all attribute numbers, it will sequentially go through the list of attribute numbers until a mask content is present.

4.11. Logging



35. Preferences - Logging

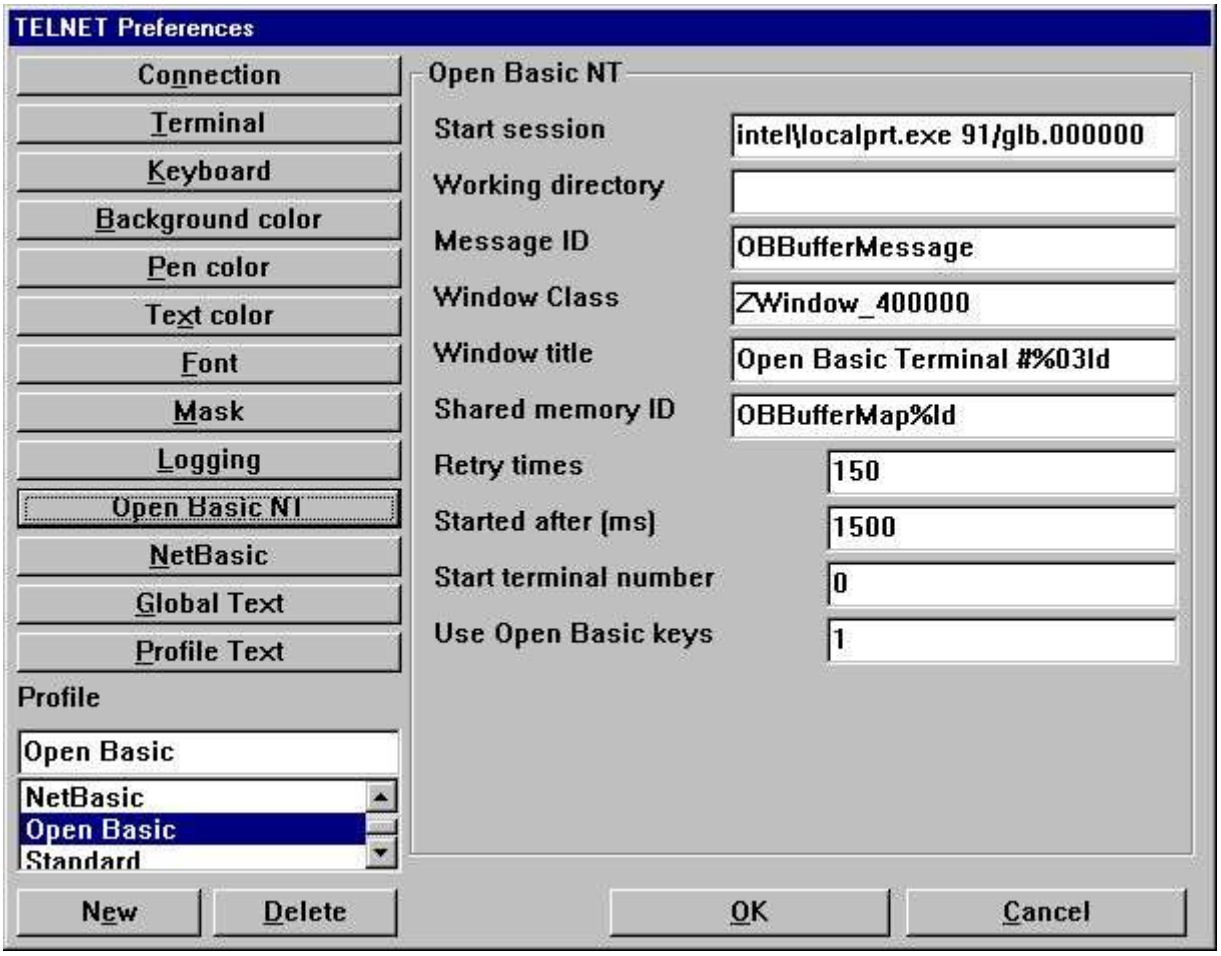
4.11.1. Log file

The field contains the full path of a log file.
If you enter the filename as

$<filename>

the leading $ sign will be replaced with the path of the loaded SWT00832.DLL or set by TelnetSettxt().
The log file is only used when the appropriate flags have been set under the connection area.
Also, the file is never reset or deleted. It must be done manually!

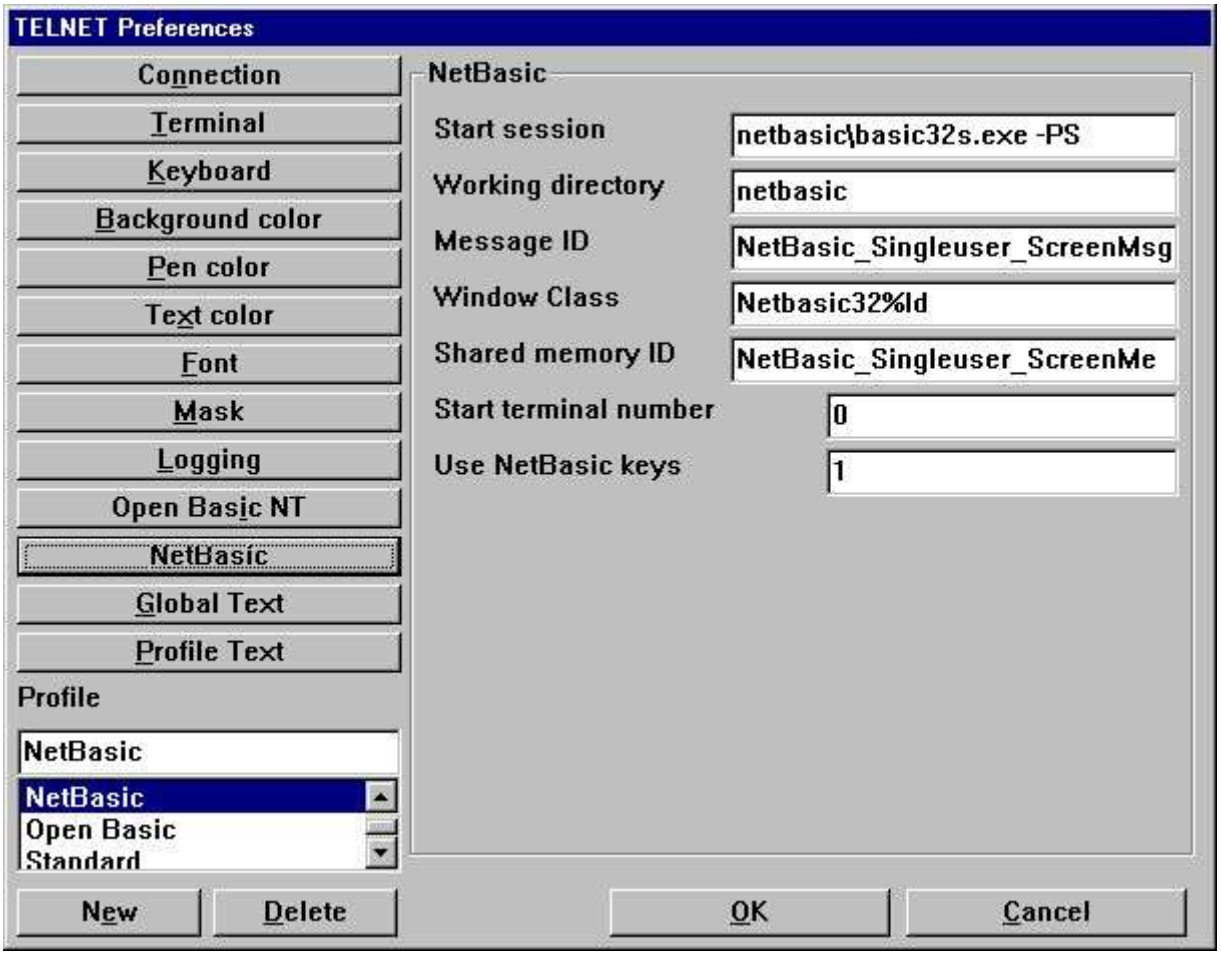
4.12. Open Basic NT



36. Preferences - Open Basic NT

Documentation will be available later.

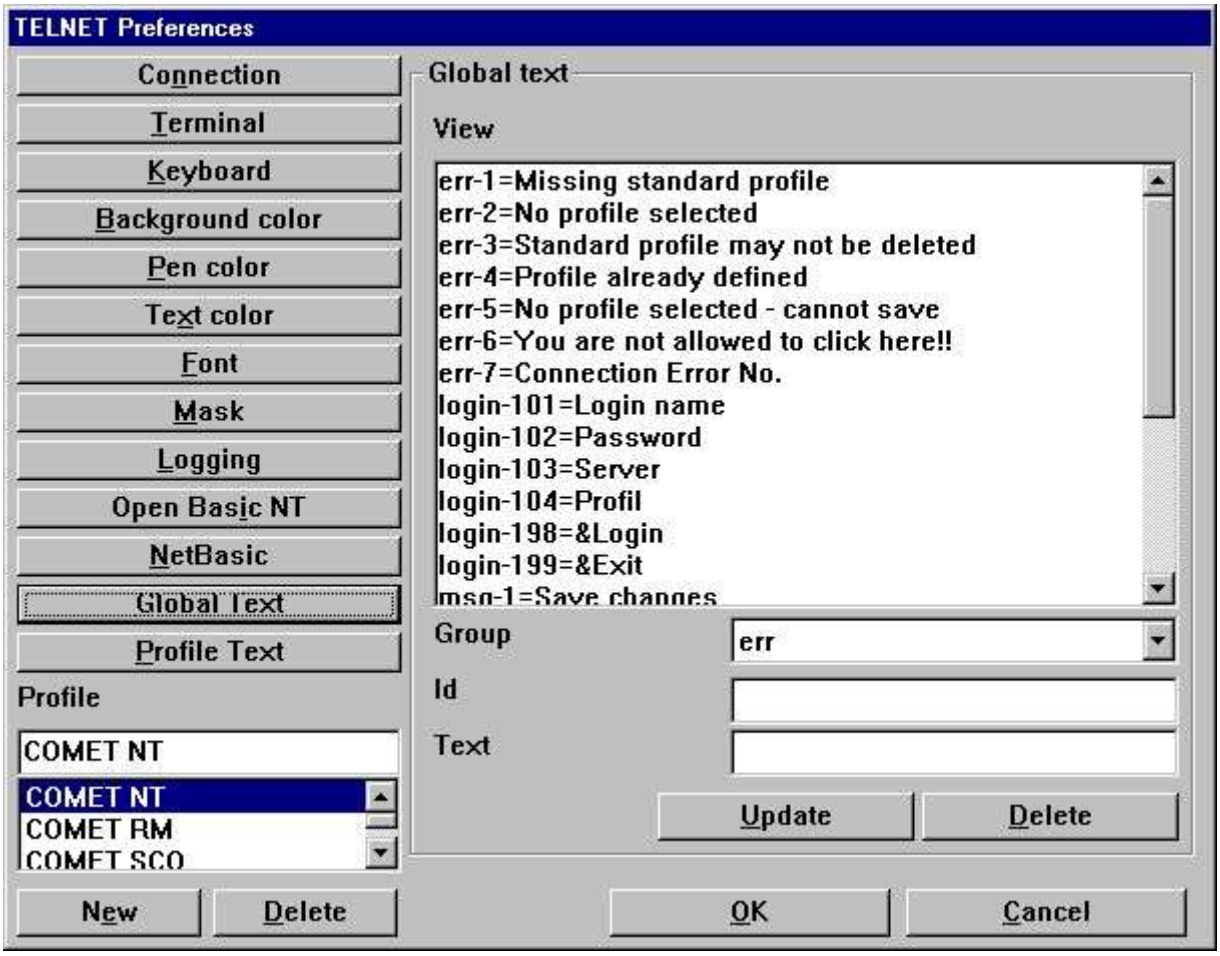
4.13. NetBasic



37. Preferences - NetBasic

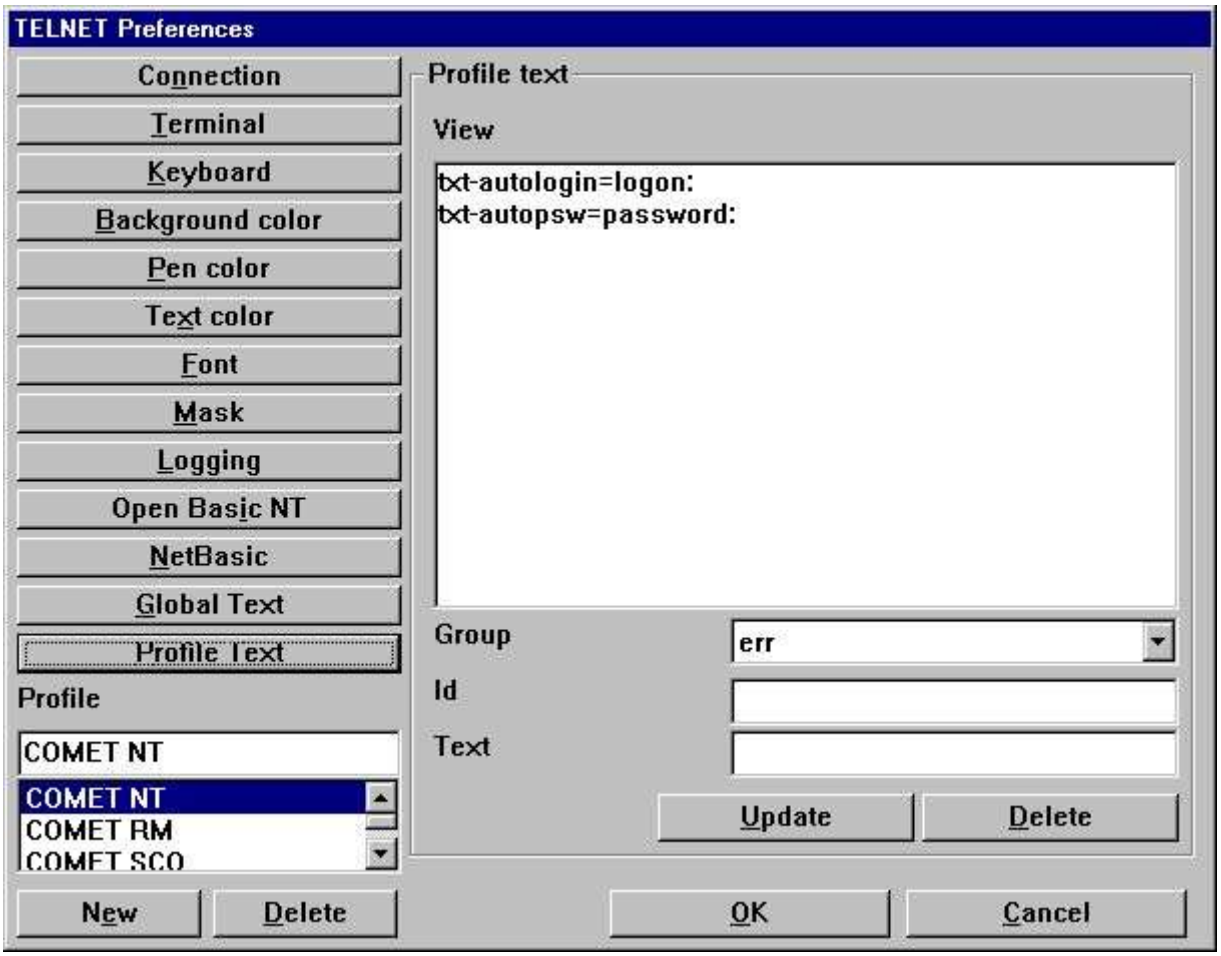
Documentation will be available later.

4.14. Global Text



38. Preferences - Global Text

4.15. Profile Text



39. Preferences - Profile Text

5. Printing

WIN-TE extends the way your character based application may print information.

5.1. Printing using scripts

There are two ways to send the print to the WIN-TE client. Via the screen, e.g. the standard output channel or via FTP, e.g. file transfer protocol.

5.1.1. Output via the screen (standard output)

For this to work you must setup a start and end escape sequence for file transfer within the terminal definition file. The default sequence within WIN-TE is setup to be

- Start equals <ESC>[5i

- End equals <ESC>4[i

This mean that the script may look something like this:

 # Script to transfer print to WIN-TE

cat startesc spoolfile endesc > /dev/tty01

where the file **startesc** contain <ESC>[5i and the file **endesc** contain <ESC>[4i.

5.1.2. Output via FTP transfer

This is not released yet! It requires a SW-Tools Print Server to run which again requires a license to SW-Tools Report Generator.

5.2. Cross Basic

The printing is implimented by an Open Interface application which may be setup with the Cross Basic device assignment. Both Windows NT and UNIX is supported as long as the Cross Basic version is V4.1 or higher.

5.2.1. Required environment variables

The Open Interface application requires some WIN-TE environment variables to be available before it can transfer a print file from the server to the client. The two variables are:

SW_WINTE_IP - Client IP address
SW_WINTE_WH - Client Application Window Identification

Because this information is only known to the client it must be able to pass this to the server. The solution is to set an option active within the WIN-TE Profile. The option is named

Set WINTE variables on server

On the server side a call must be made to a small script with the following content:

```
# WINTE.SH : (c) SW-Tools WIN-TE Set of Environment script
echo SW_WINTE_ENVIRONMENT
echo "" > ./swwintes
read SW_WINTE_VARNAME
while [ "$SW_WINTE_VARNAME" != "" ]
do
read SW_WINTE_VARVALUE
echo "$SW_WINTE_VARNAME=$SW_WINTE_VARVALUE" >> ./swwintes
echo "export $SW_WINTE_VARNAME" >> ./swwintes
read SW_WINTE_VARNAME
done
chmod +x ./swwintes
. ./swwintes
```

You may find this content in the file **WINTE.SH** in the **c:\swtools\winte\bin** folder.
On Windows NT you may copy the file to a desired location, e.g.

c:\swtools\winte\bin\winte.sh ==> /X.BASIC/0/winte.sh

and amend the following line to the **/X.BASIC/0/profile.ksh**

. winte.sh

On UNIX systems you may copy the file to a desired location, e.g.

c:\swtools\winte\bin\winte.sh ==> /X.BASIC/0/winte.sh

and amend the following line to the **/X.BASIC/0/.profile**

. winte.sh

Please remember the leading dot and space before winte.sh. Also, the call to winte.sh must be before the start of tmenu!

5.2.2. SW-Tools WIN-TE Print Server

The WIN-TE Print Server is installed when you install SW-Tools WIN-TE. It is a service that runs on the client PC or on the server where NET-TRIO is installed. It will retrieve the print from the character based application and forward it to the WIN-TE client.
You may right click on the icon in the taskbar to see what has been transfered from your application.
No print can be retrieved if this service is not running.

5.2.3. Device assignment

Before you may setup the driver and device assignment you must copy the file

c:\swtools\winte\bin\swoi.nt - Windows NT
c:\swtools\winte\bin\swoi.rmx - RM UNIX

to your Cross Basic system. For example to the

/X.BASIC/0/swoi.exe - Windows NT
/X.BASIC/0/swoi - RM UNIX

From the Cross Basic superuser menu you select

1. Manager functions - Utilities - Device assignment
2. Select - Driver definition
3. Edit - Create

The creation of the driver definition may be

Driver name: $SWOI
Device type: Open interface
Host/program: /X.BASIC/0/swoi.exe
Service/parameter0: swoi
Parameters: /def=/X.BASIC/0/xbasic.oi

Now to setup a printer you may select menu

1. Select - Device assignment - User - Foreground port
2. Edit - Create

The creation of the device definition may be

Logical name: $LPT
Device type: Printer and other interfaces ($SWOI)

and save it by selecting OK.

5.2.4. SWOI application parameters

/dbg=<debug filename>
/def=<definition filename>
/dir=<spool directory>
/ftp=<fixed ipaddress>|<hostname>|$<ENVIRONMENT VARIABLE>
/usr=<ftp username>
/psw=<ftp password>
/out=<ftp local output directory>
/por=<ftp portnumber>
/tim=<ftp timeout in seconds>
/c70=0|1

5.3. Codetable conversion and filter

5.3.1. Codetable

You may inform the WIN-TE client to perform a code conversion on the file by adding the following line to the definition file.

//CODETABLE=013

Please refer to the codetables in SW-Tools Data Dictionary. Here we will only list some of the available codetable numbers:

013 - OemToAnsi conversion
044 - BASIC Parity English
049 - BASIC Parity German

5.3.2. Filter

No documentation yet!

5.4. Extended print control

//PRINT=cop=2,15
We now have two copy of our print on printer 15
//PRINT=prt=2,17
This comes out on the default printer
even if we have opened printer channel 2 which is the
SW-Tools printer number 17.
//PRINT=>2
But this line comes out on the printer channel 2
//PRINT=>1
Now we are back on printer 1

5.5. Archiving

You may extend the printing with archiving by adding the following line in the definition file:

//PRINT=log=pip=c:\archive.zip

But the simplest way is to activate archiving on the printer within WIN-TE.

5.6. Forms

Forms are easely controlled in the SW-Tools printer setup within WIN-TE.

6. Functions

The collection of TELNET functions are installed from a Windows DLL (Dynamic Linked Library). All the functions are described and implemented as sub functions in TRIO in the 15.xxx file, where xxx is the language code, located in the TRIO installation directory.
The on-line description of the TELNET function may be viewed directly from the database window. Select the file names 'TELNET functions', which is file id 15.

6.1. **TelnetAllocate** - Allocate Telnet object

#1 = TelnetAllocate() /* *Allocate new telnet object*

6.2. **TelnetFree** - Free telnet object

See TelnetAllocate

6.3. **TelnetConnect** - Connect Telnet

This function makes the actual connection with the host. It will initialize the connection according to the profile given in *par2*.
If *par3* is "" it will be taken from the profile.
If *par4* is "" it will be taken from the profile. It may also be the actual portnumber plus 1000000 which will make sure that no login dialog is used even if set on the profile.

TelnetDisconnect(#1) /* *Disconnect from server*

6.4. **TelnetDisconnect** - Disconnect Telnet

This function will disconnect an already connected session.

See TelnetConnect

6.5. **TelnetPut** - Put text to Telnet

This function will send a text of a given length to simulate input.

TelnetPut(#1,"ls\n",3) */* *perform command ls (\n = CR)*

6.6. TelnetNotify - Notify when text received

This function will set a wait buffer text. If and when the terminal emulator receives this text it will send a notification to the IQ program. In order to process such a notification the IQ program place its calculations under the section "Telnet Object" + "Text 1 arrived" .. "Text 16 arrived".

Telnetput(#1,"root\n", 5) /* *Automatically login as root*

6.7. **TelnetKey** - Put key sequence to Telnet

This function sends key sequences as defined in the used keyboard table. The sequence given in *par2* must a reference to the internal key number. The reference may be preceeded by 's', 'c' and 'a' where
s = Shift c = Ctrl a = Alt
to enable sequences like
sc49 (Shift + Ctrl + '1')

TelnetKey(#1,"027.013") */* *Keys ESC followed by CR send to telnet*

6.8. **TelnetGetnum** - Get Telnet numeric variable

This function returns a numeric value known in the telnet session according to the *par2*.

#2 = TelnetGetnum(#1,8) */* *Did user click within telnet client area*

6.9. **TelnetGettxt** - Get Telnet text variable

If *par2* is less than zero it will represent the line number from which the entire line content should be retrieved.

TelnetGettxt(#1,4,#2) /* *Retrieves the text the user have clicked on*

6.10. TelnetSaveScreen - Save Telnet screen content

TelnetSaveScreen(#1)

6.11. **TelnetActivate** - Activate focus on Telnet

This function makes sure that focus is on the telnet object. This is useful when window have stolen the focus.

TelnetActivate(#1)

6.12. **TelnetPref** - Preferences

This function displays the preference dialog for the telnet object. You are able to create, amend and delete profiles.

TelnetPref(#1)

6.13. **TelnetSetnum** - Set Telnet numeric variable

Thsi function allows you to set a numeric variable within the telnet object.

TelnetSetnum(#1,0,1) /* *Start mark of screen content*

6.14. **TelnetSettxt** - Set Telnet text variable

6.15. **TelnetFileFilter** - Filter file according to printer filter

The function will filter the file content according to the printer filter setup on the profile.

TelnetFileFilter(#1,"c:/sw000001.txt")

Figure list

1. Application appearance before using WIN-TE 4
2. Application appearance after using WIN-TE 5
3. The WIN-TE applications 13
4. Step 1 - Login to the system 15
5. Step 2 - Started Debtors Master data maintenance 16
6. Step 3 - Definition of a unique mask number 17
7. Step 4 - Automatic recognition of (dotted) input fields 18
8. Step 5.1 - Defining a pop-up menu for 3-letter commands 19
9. Step 5.2 - Creating a record for a 3-letter command 20
10. Step 5.3 - The final pop-up menu 21
11. Step 6.1 - Defining an IQ query 24
12. Step 6.2 - The query window 24
13. Step 6.3 - Saving the query 25
14. Step 6.4 - Calculations to return value to WIN-TE 26
15. Step 6.5 - Action for click on debtor number field 27
16. Step 6.6 - Finding and selecting the debtor 28
17. Step 6.7 - Application displays debtor data 29
18. Step 7 - Defining input fields manually 30
19. Step 8 - Defining display fields manually 31
20. Step 9.1 - Defining a button field 32
21. Step 9.2 - Defining a button field text 33
22. Step 9.3 - Action for click on button 34
23. Step 9.4 - IQ Calculations required to retrieve values from WIN-TE 35
24. Step 9.5 - Application enhanced with button 36
25. Step 9.6 - Action for click on button with copy of text 37
26. Step 9.7 - Write started and debtor information inserted 38
27. Preferences 40
28. Preferences - Terminal 53
29. Preferences - Keyboard 59
30. Preferences - Background color 65
31. Preferences - Pen color 66
32. Preferences - Text color 67
33. Preferences - Font 68
34. Preferences - Mask 70
35. Preferences - Logging 78
36. Preferences - Open Basic NT 80
37. Preferences - NetBasic 81
38. Preferences - Global Text 82
39. Preferences - Profile Text 83

Index

B
button
.5;18;19;26;29;30;31;32;33;34;35;36;49;51;67;115
C
COMET 10;14;23
D
DATAMASTER..... 12;16;18;19;26;31
display 5;30;31;51;115
I
input. 5;17;22;26;29;30;50;51;63;103;115
IQ
.9;12;14;23;24;26;28;32;33;34;36;39;51;104;115
N
NetBasic..... 6;14;80;115
NET-TRIO 9;89
P
pop-up 18;19;20;28;115
T
Telnet
51;57;99;101;102;103;104;105;106;107;109;110;112;113
TRIO 5;7;9;22;98
W
WIN-TE
1;3;4;5;6;11;12;24;25;27;31;33;34;36;83;84;85;88;89;93;96;97;115